



Typical partition between firmware and hardware

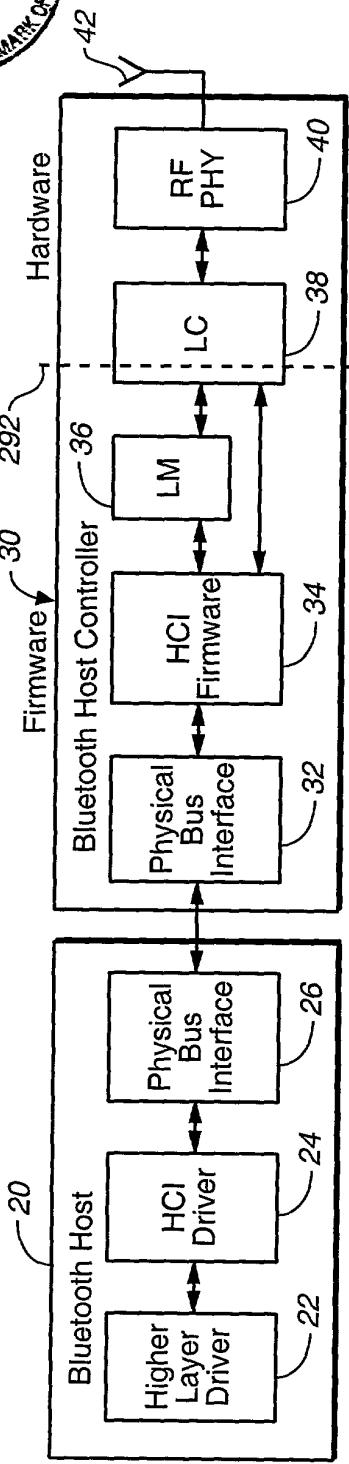


FIG. 1
 (PRIOR ART)

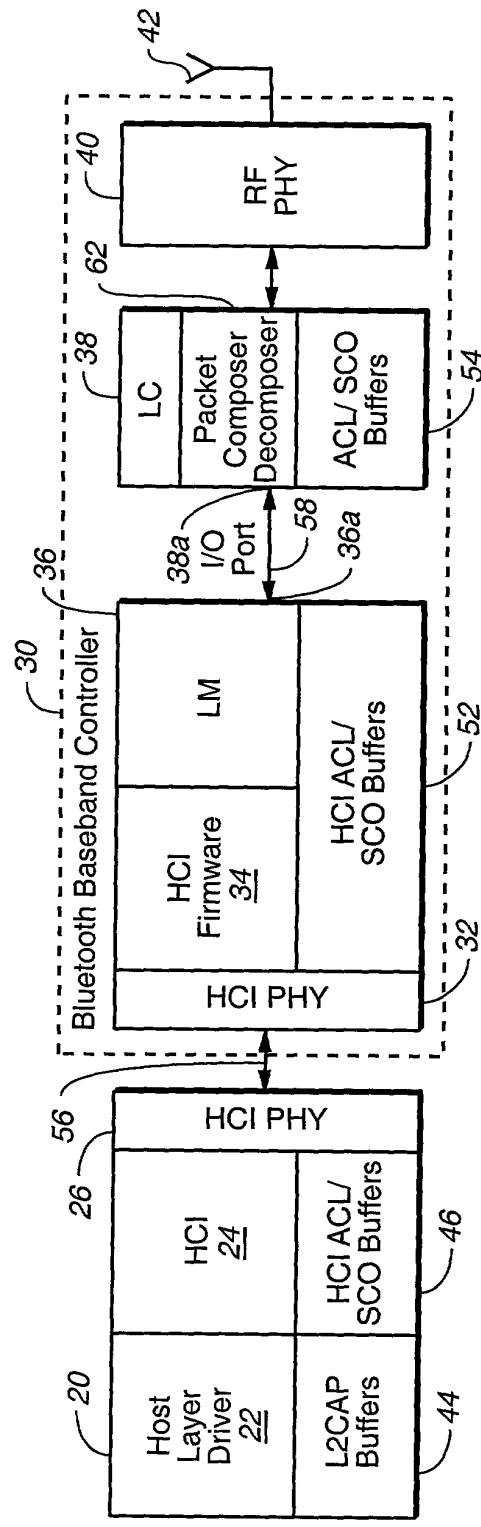


FIG. 2
 (PRIOR ART)



HCI ACL Data Packet

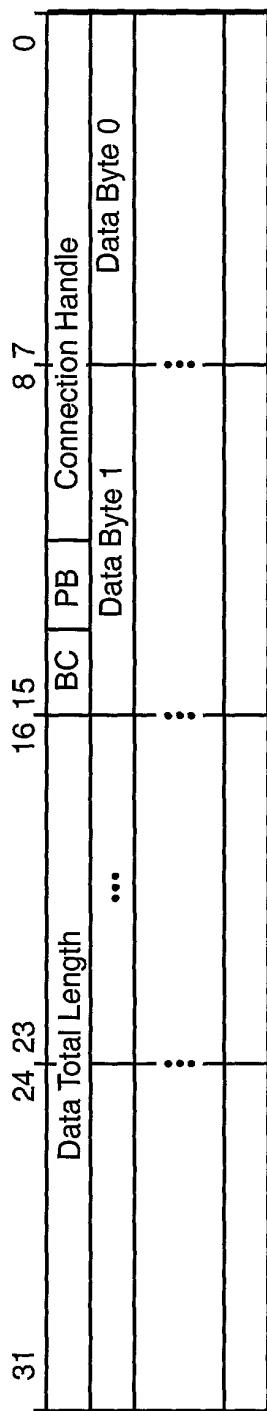


FIG._3A
(PRIOR ART)

HCI SCO Data Packet

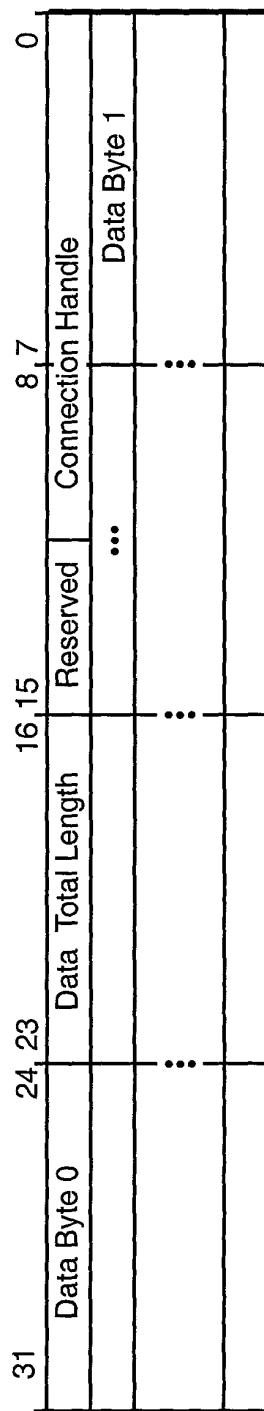


FIG._3B
(PRIOR ART)



Data Flow and Buffer Scheme in the Bluetooth Host Controller

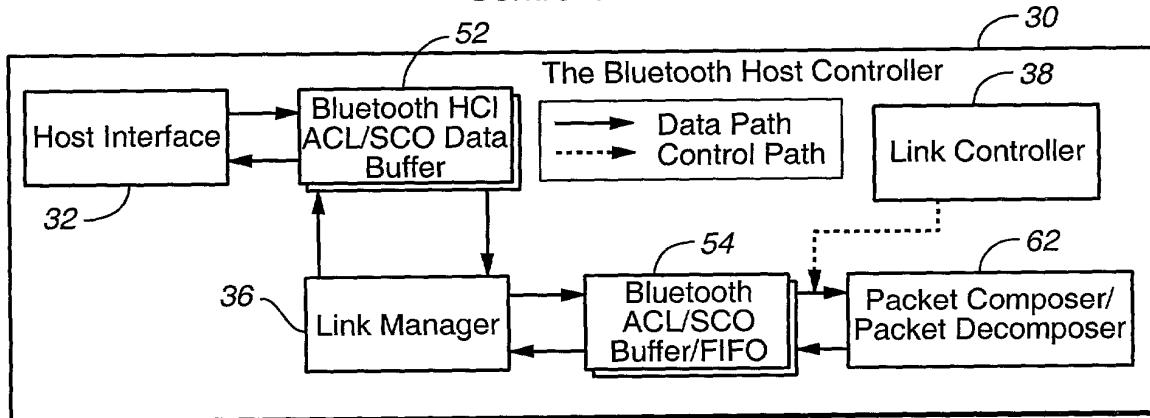


FIG._4
 (PRIOR ART)

Dual Buffer Scheme For ACL Packet Transmission

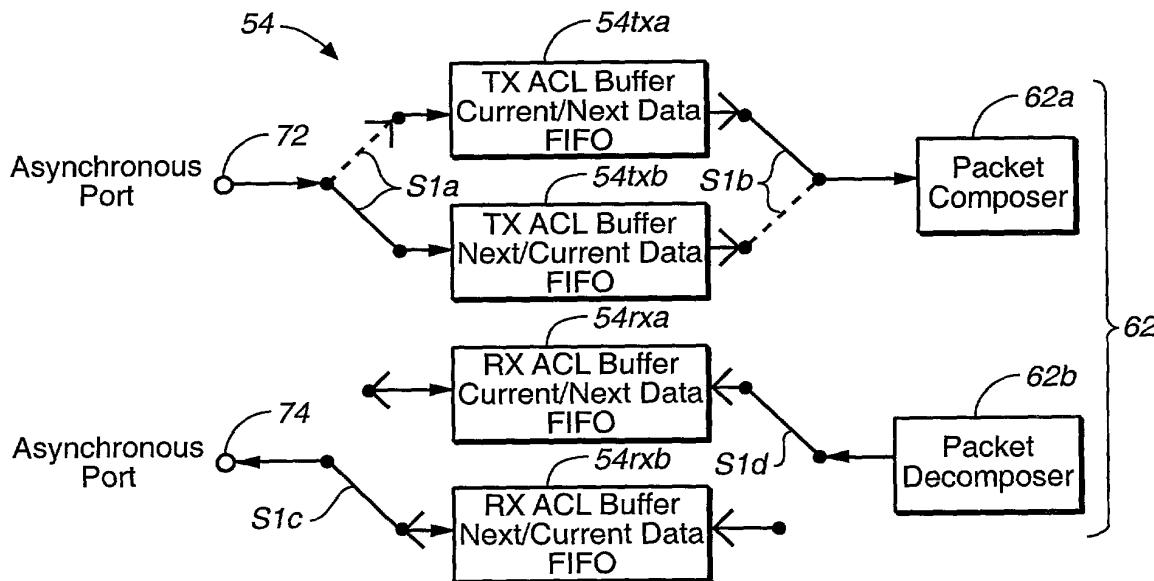


FIG._5A
 (PRIOR ART)

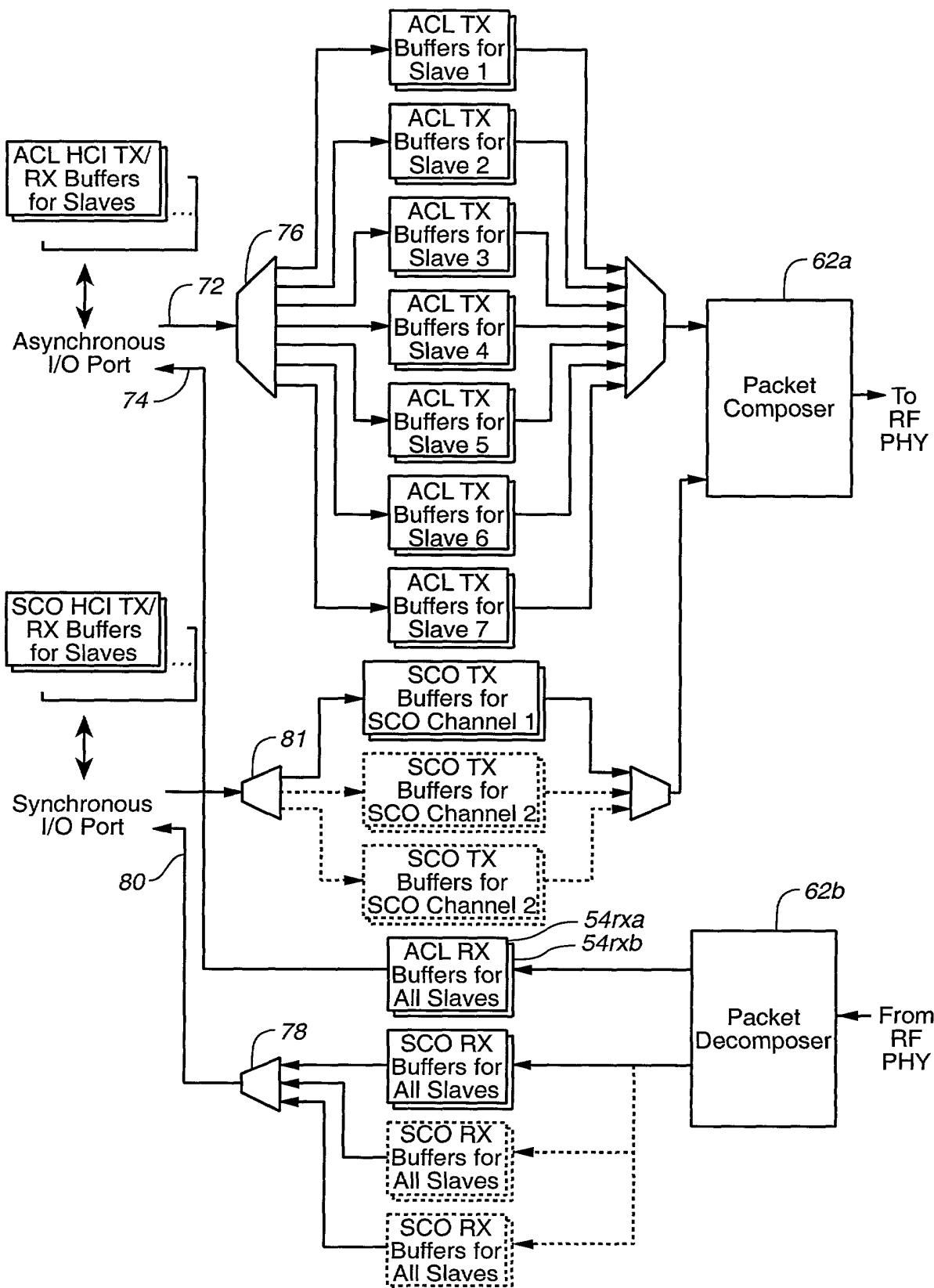
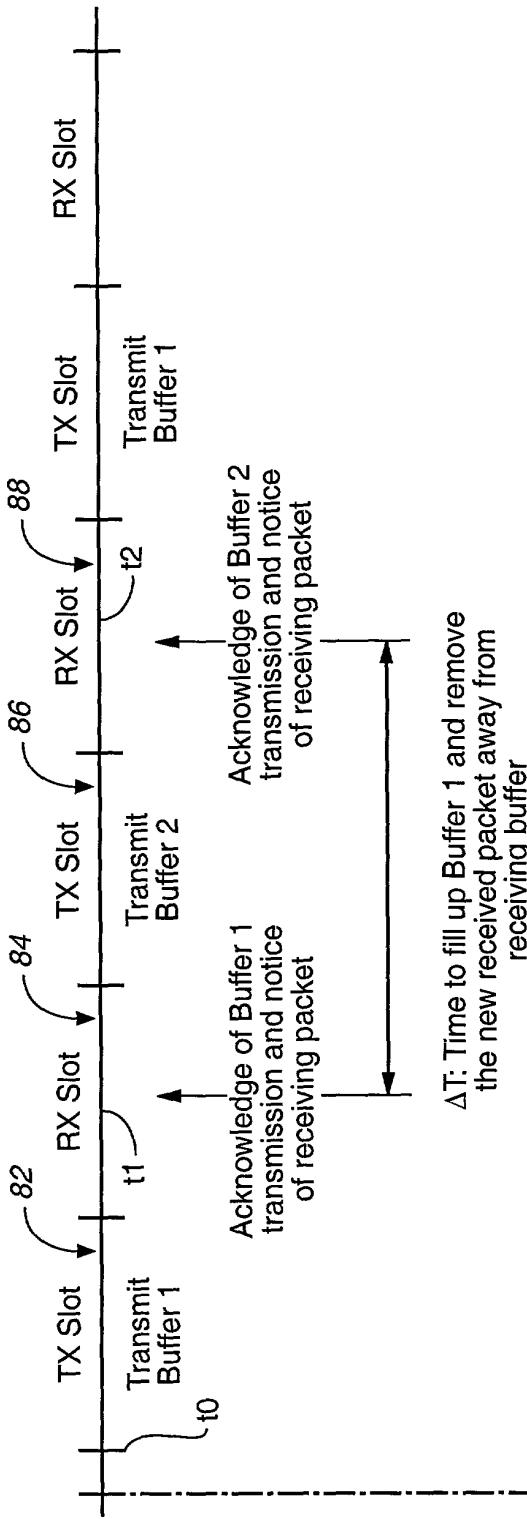


FIG.-5B (PRIOR ART)



The worst case timing (ΔT) for the LM to load a TX buffer and unload a RX buffer



ΔT : Time to fill up Buffer 1 and remove the new received packet away from receiving buffer

FIG.-6
 (PRIOR ART)



The Packet Generator accesses the HCI ACL / SCO buffers directly

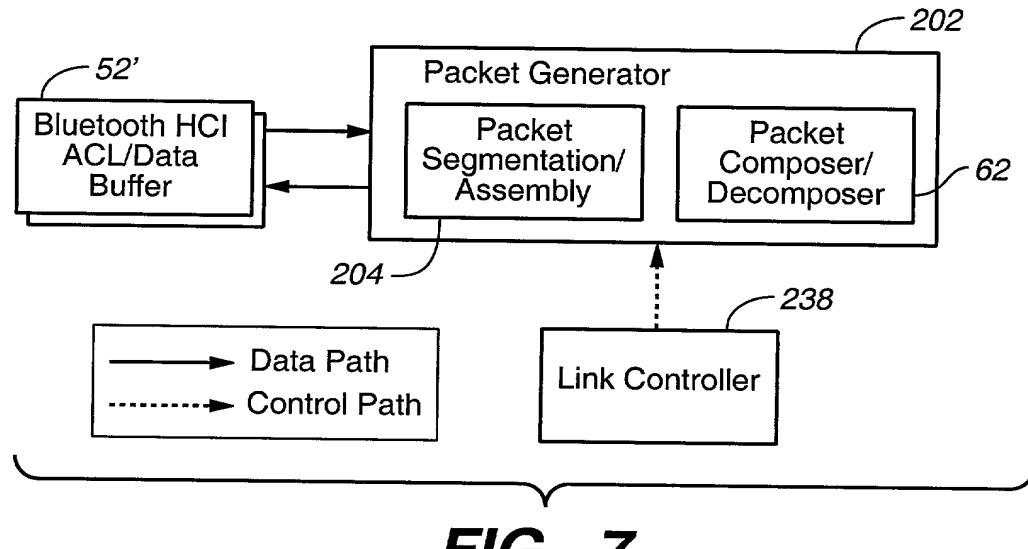


FIG._7

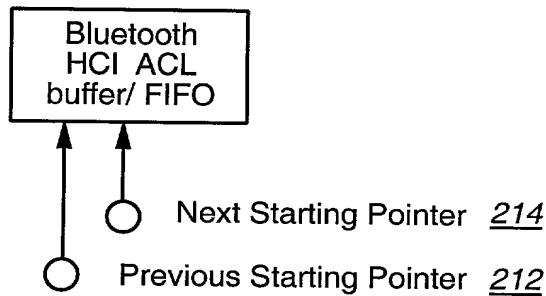


FIG._8

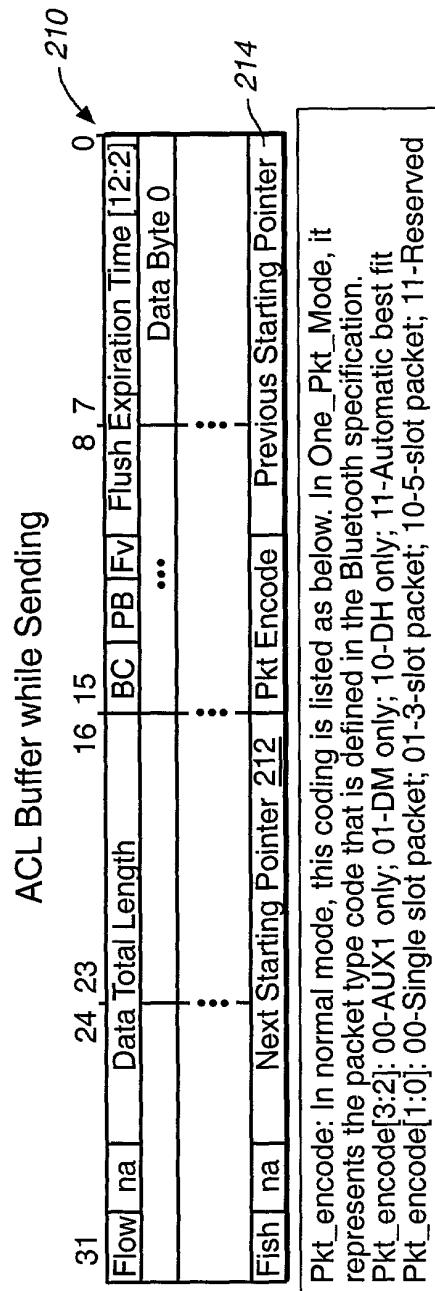


FIG. - 9A

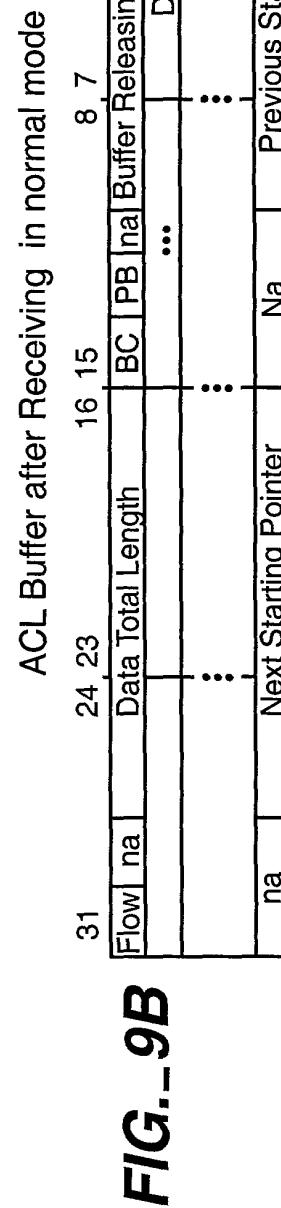
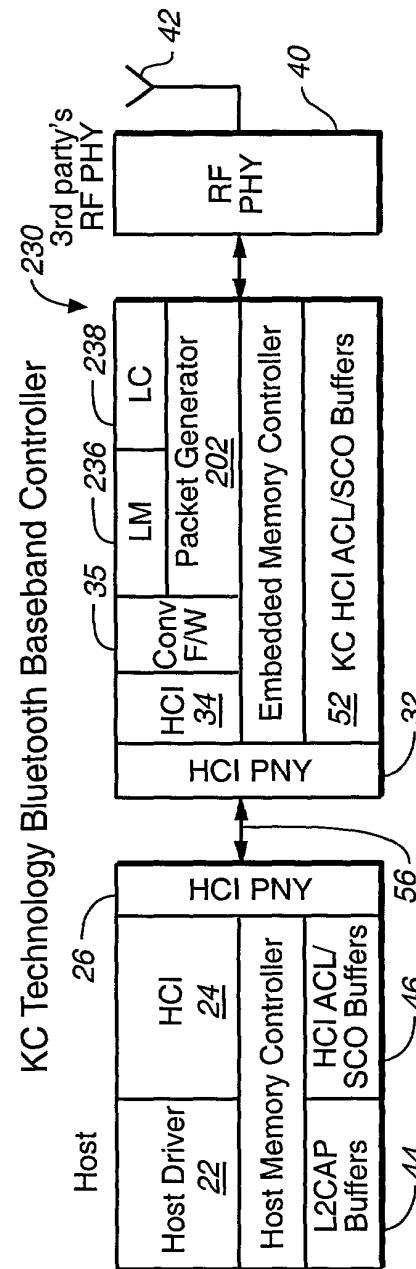


FIG. 10





ACL transmission control flow of KC Technology's partition

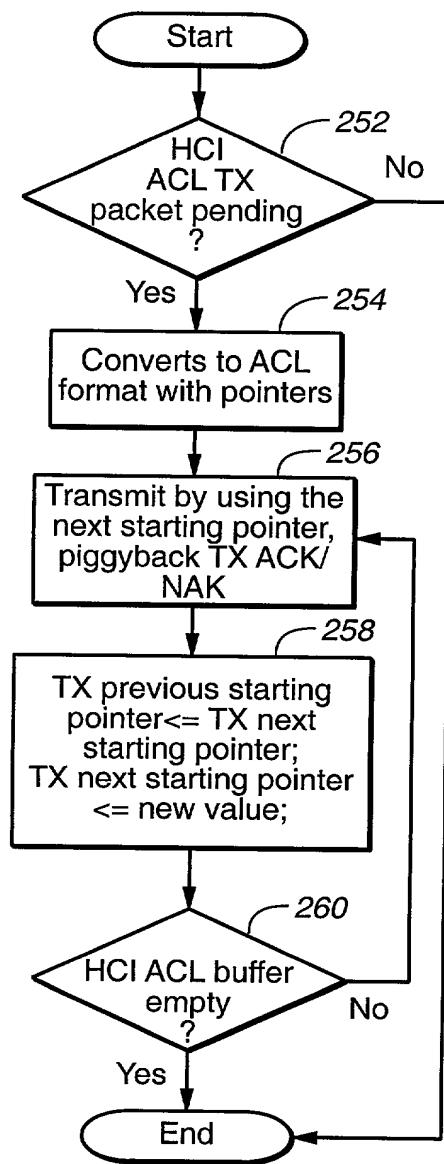


FIG._11

ACL receiving control flow of KC Technology's partition

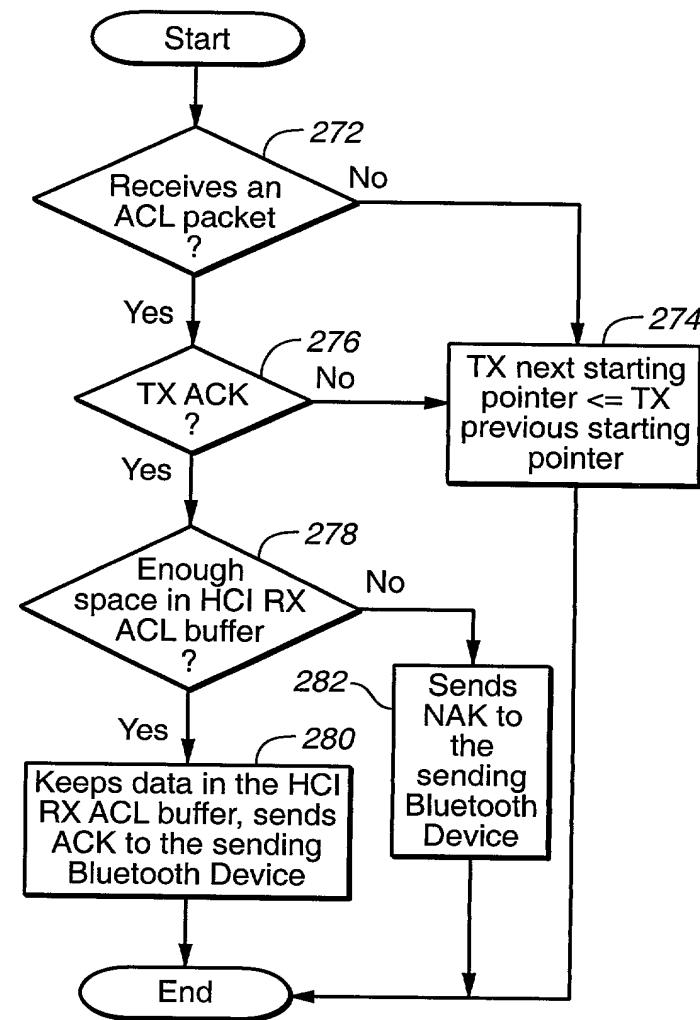


FIG._12



KC Technology's Partition between firmware and hardware

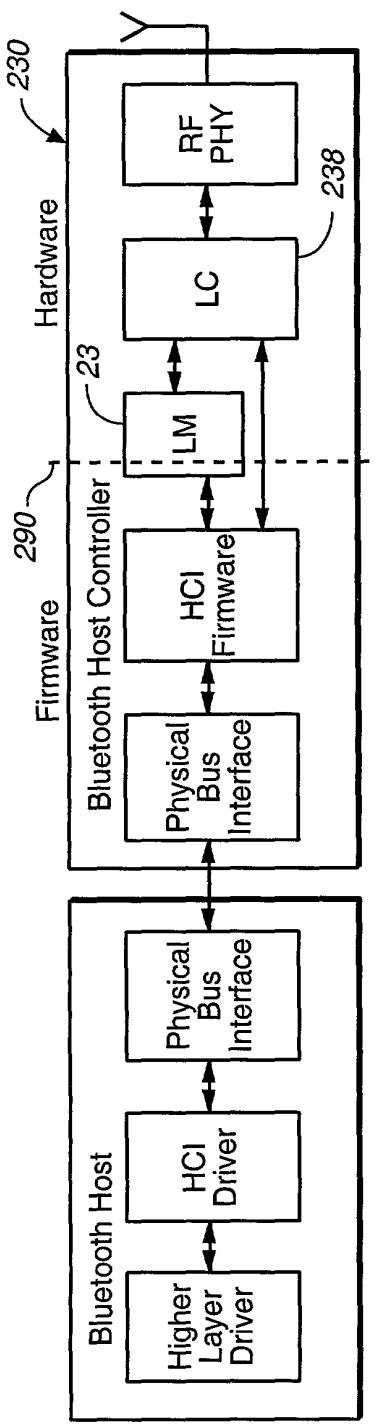


FIG._ 13

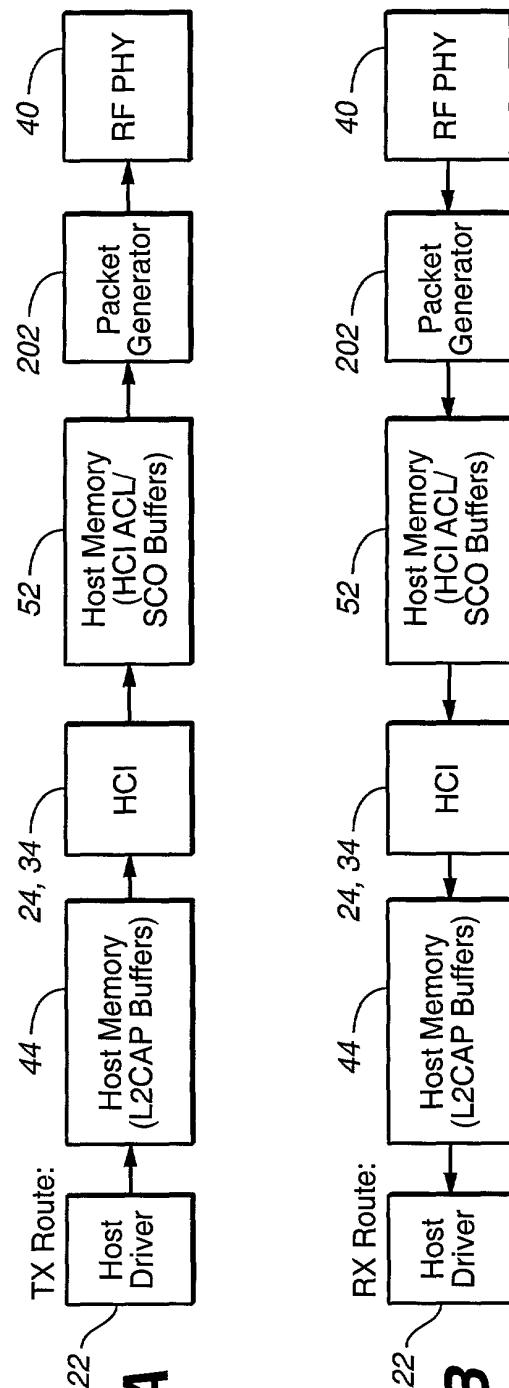


FIG._ 14A

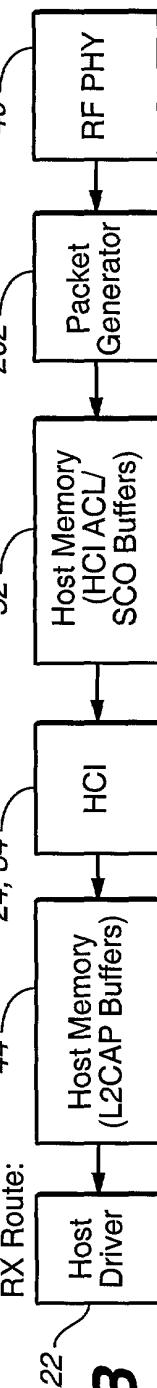


FIG._ 14B



Type A-2: Host / KC Technology Bluetooth BaseBand Controller

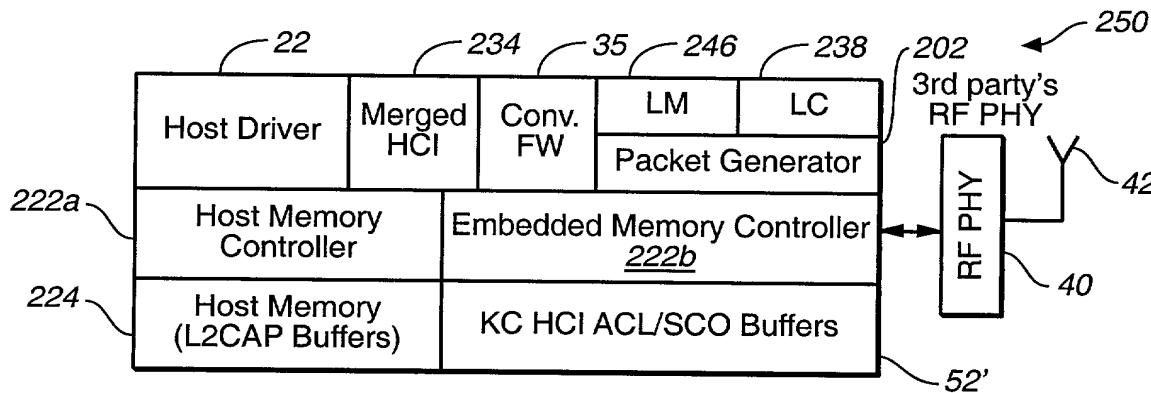


FIG. 15A

Type A-2: Host / KC Technology Bluetooth BaseBand Controller

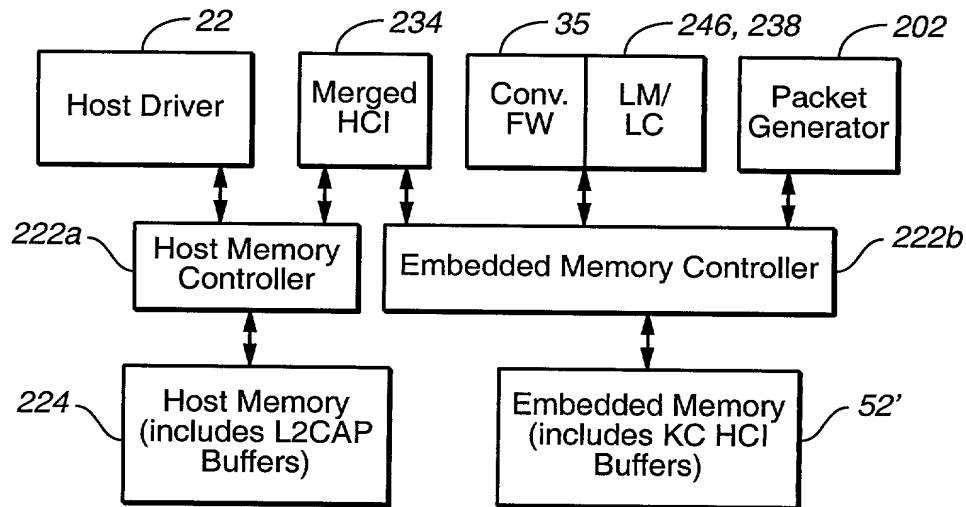
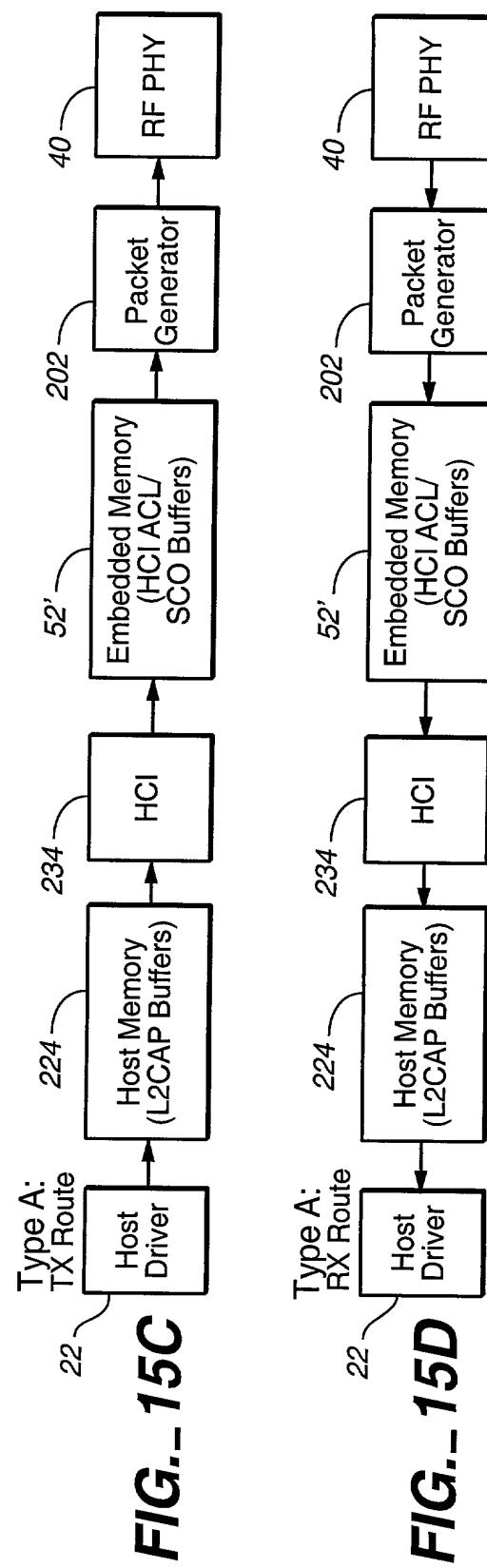


FIG. 15B





12 / 45

Type A-1: Host / Bluetooth BaseBand Controller

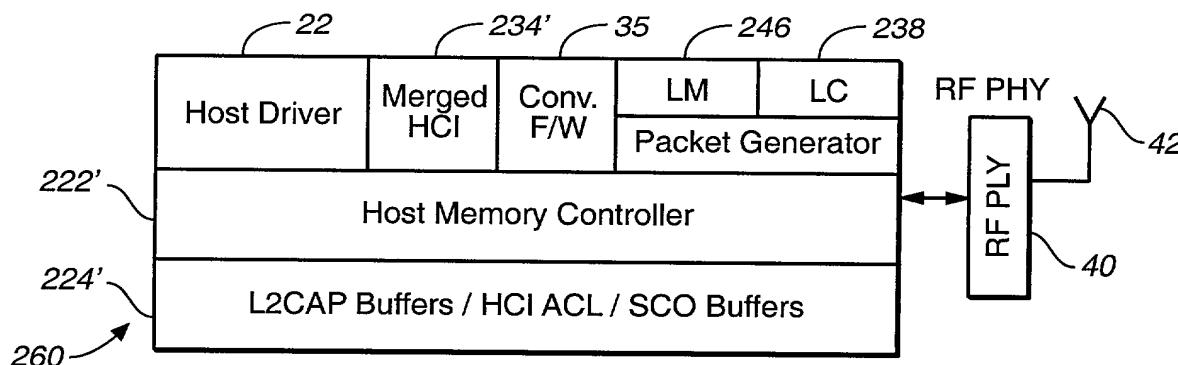


FIG._ 16A

TYPE A-1: Host / Bluetooth BaseBand Controller

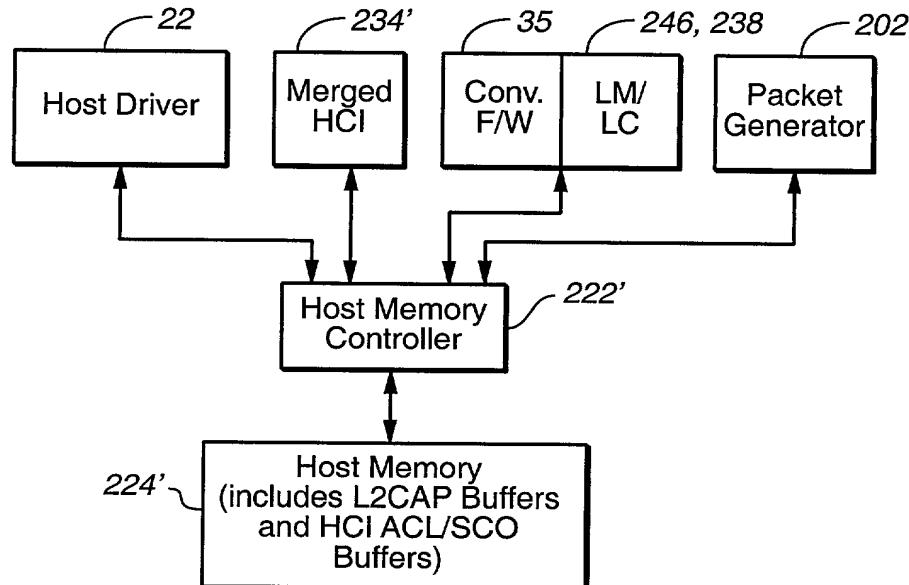


FIG._ 16B

Type B: Host / Bluetooth BaseBand Controller

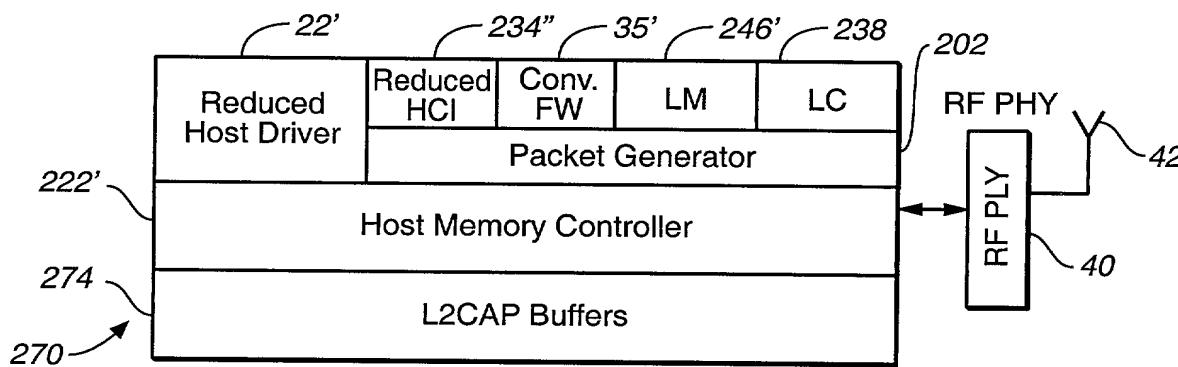


FIG._ 17A



13 / 45

Type B: Host / Bluetooth BaseBand Controller

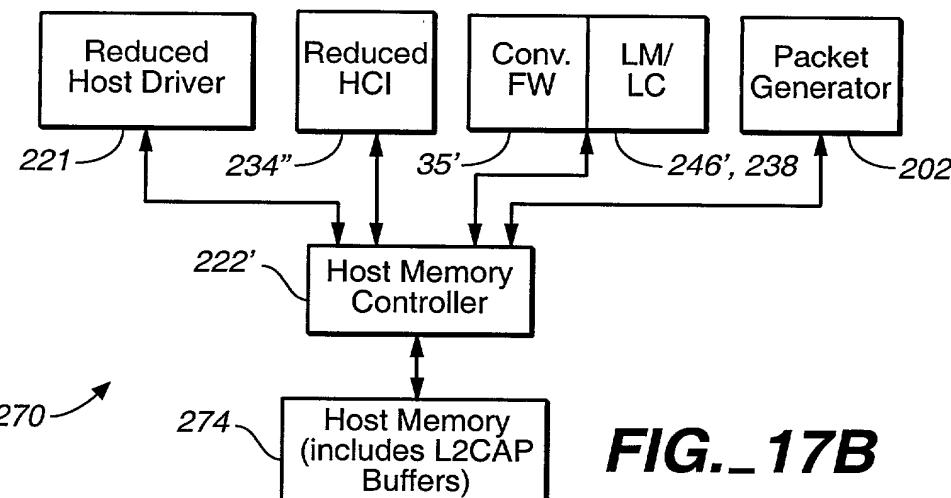


FIG._ 17B

Type B: TX Route

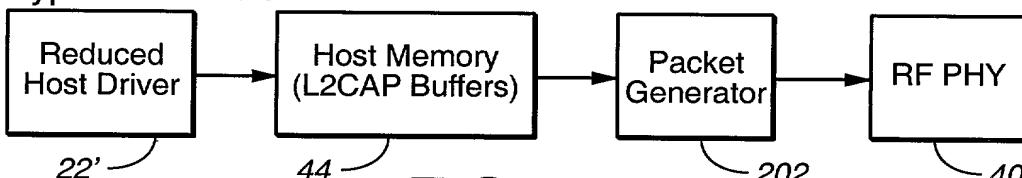


FIG._ 17C

Type B: RX Route

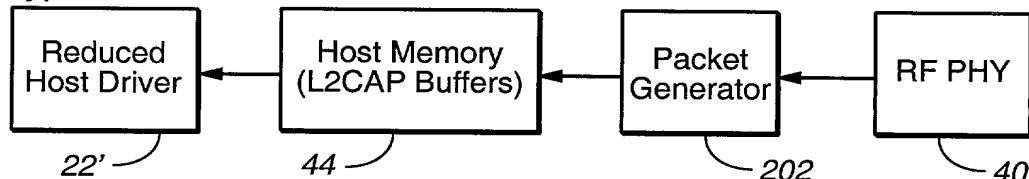


FIG._ 17D

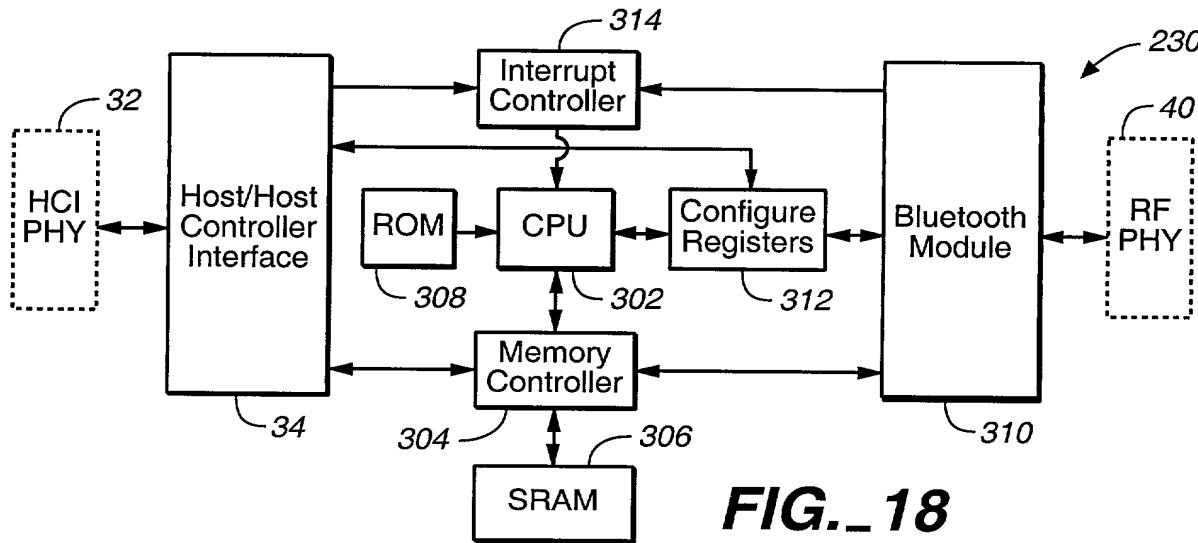
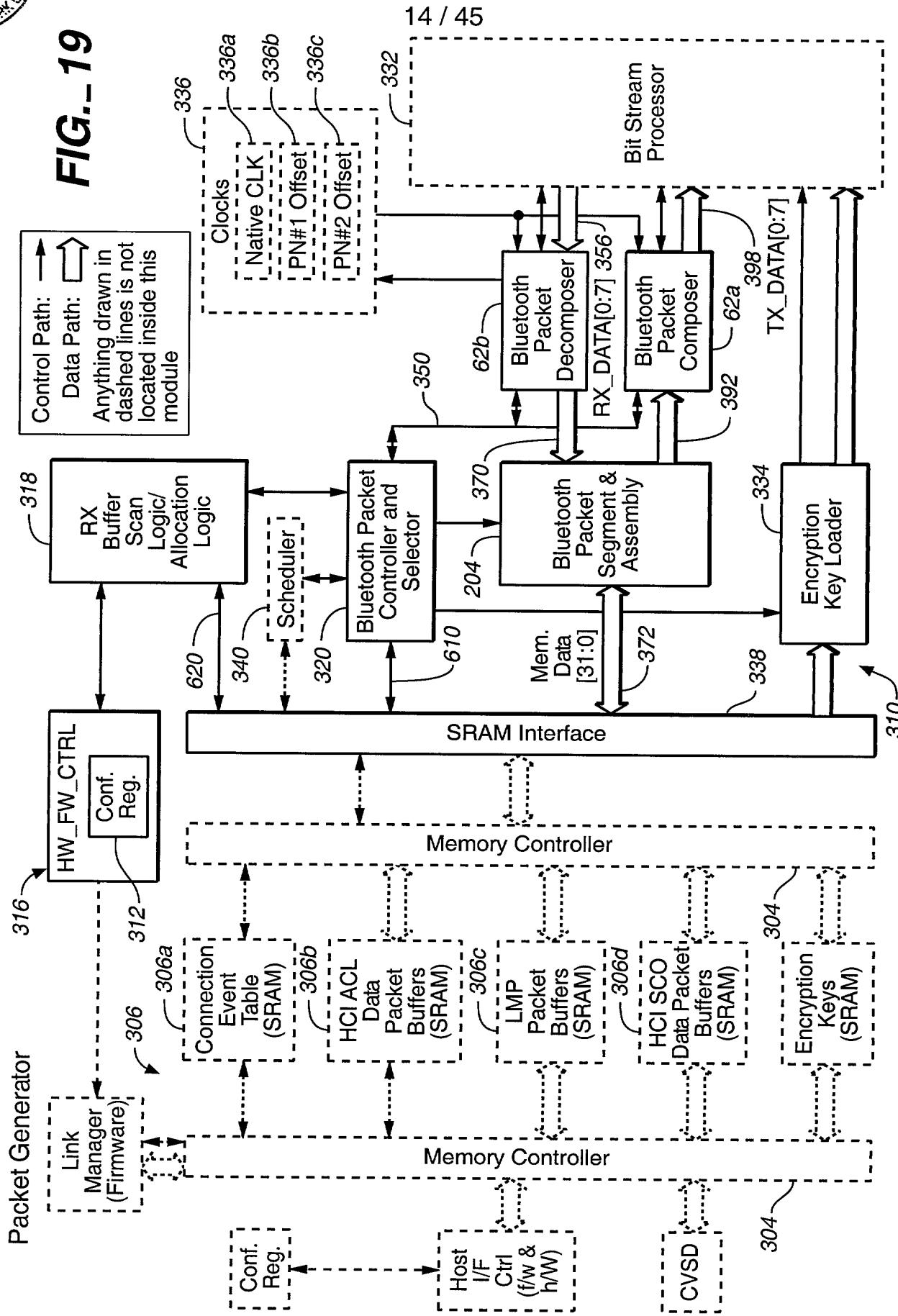


FIG._ 18



FIG.-19





Outgoing FHS Packet storage format:

31							
24 23 16 15 8 7 0							
na				0 1 1 1 0 BC 0 0		na	
UAP [1:0]	SP [1:0]	SR [1:0]	na	LAP			
CLASS[9:0]				NAPI[15:0]		UAP[7:2]	
na				AM_ADR [2:0]	CLASS[23:10]		
na				PSM[2:0]		na	

FIG._20

Incoming FHS Packet storage format:

31							
24 23 16 15 8 7 0							
na				0 1 1 1 0 BC 0 0 1		na	
UAP [1:0]	SP [1:0]	SR [1:0]	na	LAP			
CLASS[9:0]				NAPI[15:0]		UAP[7:2]	
CLK[16:2]				AM_ADR [2:0]	CLASS[23:10]		
na				PSM[2:0]		CLK[27:17]	

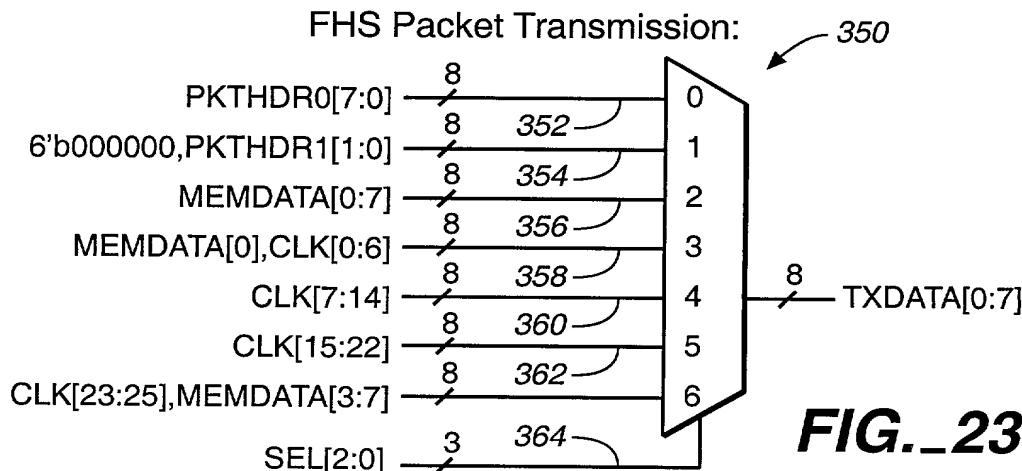
FIG._21



Data Byte Sending Sequence in FHS packet:

Byte# & name	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7							
1:Packet Header 0	amadr0	amadr2	amadr2	pkttp0	pkttp1	pkttp2	pkttp3	flow							
2:Packet Header 1	arqn	seqn	-	-	-	-	-	-							
3:FHS 0	lap0	lap1	lap2	lap3	lap4	lap5	lap6	lap7							
4:FHS 1	lap8	lap9	lap10	lap11	lap12	lap13	lap14	lap15							
5:FHS 2	lap16	lap17	lap18	lap19	lap20	lap21	lap22	lap23							
6:FHS 3	tbd0	tbd1	sr0	sr1	sp0	sp1	uap0	uap1							
7:FHS 4	uap2	uap3	uap4	uap5	uap6	uap7	nap0	nap1							
8:FHS 5	nap2	nap3	nap4	nap5	nap6	nap7	nap8	nap9							
9:FHS 6	nap10	nap11	nap12	nap13	nap14	nap15	clss0	clss1							
10:FHS 7	clss2	clss3	clss4	clss5	clss6	clss7	clss8	clss9							
11:FHS 8	clss10	clss11	clss12	clss13	clss14	clss15	clss16	clss17							
12:FHS 9	clss18	clss19	clss20	clss21	clss22	clss23	amad0	amad1							
13:FHS 10	amad2														
14:FHS 11															
15:FHS 12															
16:FHS 13															
				pgscn0	pgscn1	pgscn2	-	-							

FIG._22





17 / 45

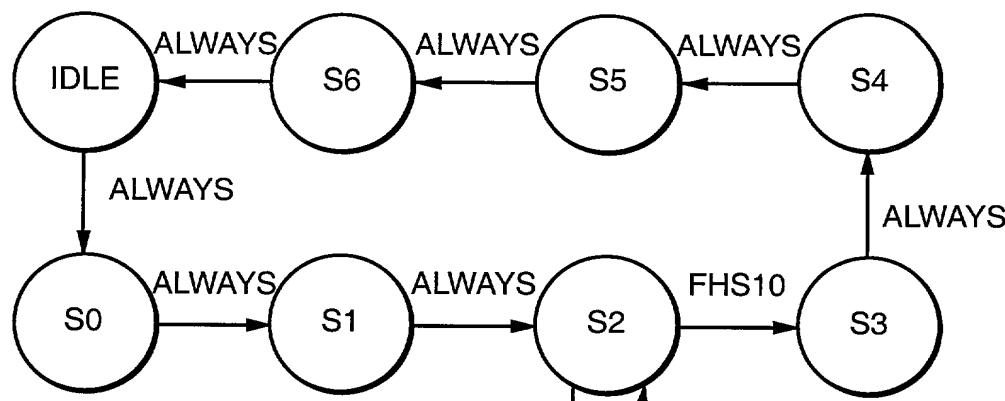


FIG._24A

!FHS10

FIG._24B

State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	2
S3	3
S4	4
S5	5
S6	6

Data Byte Sending Sequence in DM1 packet:

Byte# &
name

1:Packet
Header 0

2:Packet
Header 1

3:Payload
Header 0

4:ACL

data 1

5:ACL

data 2

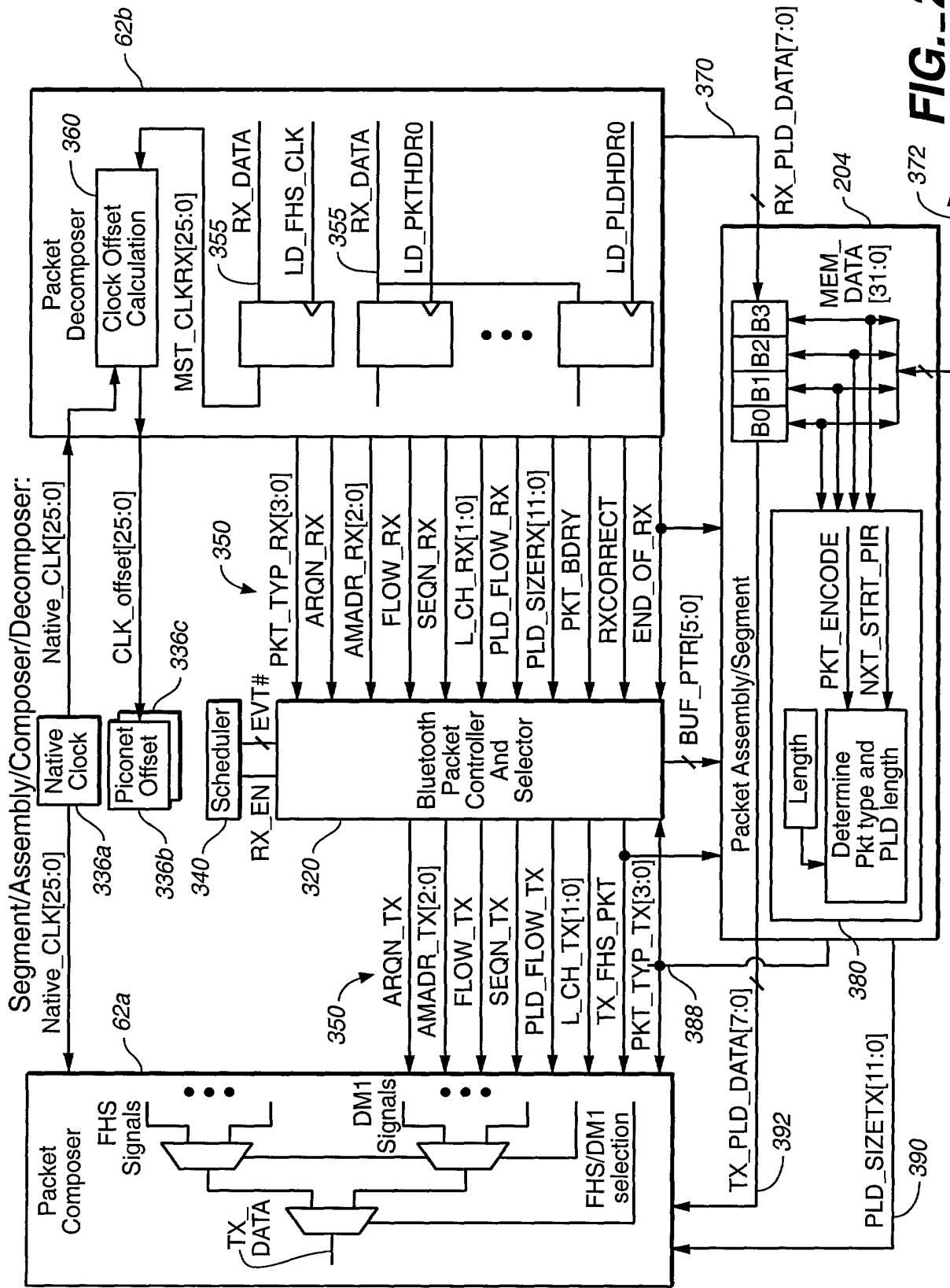
...

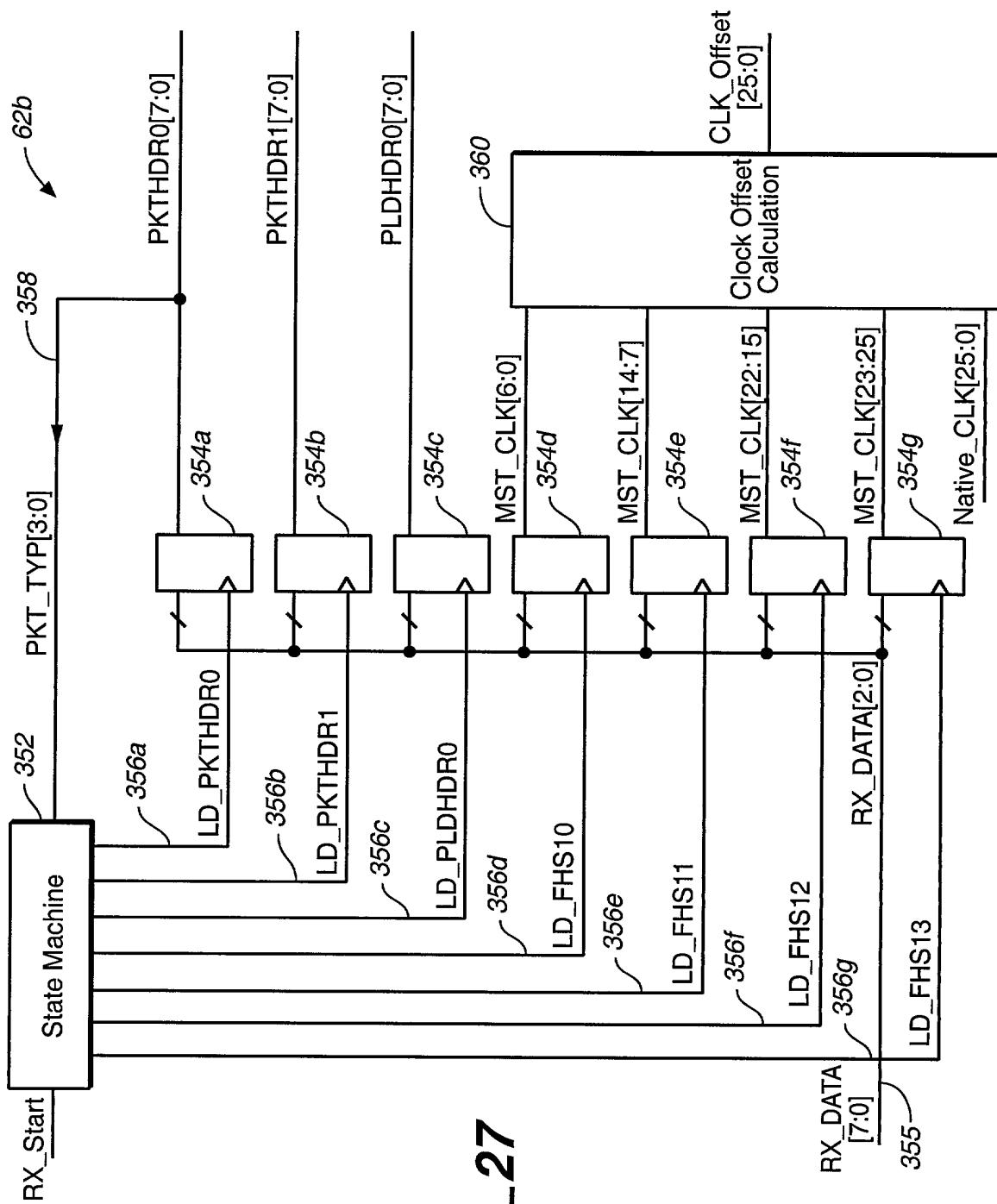
n+3:ACL
data n

Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
amadr0	amadr1	amadr2	pktp0	pktp1	pktp2	pktp3	flow
arqn	seqn	-	-	-	-	-	-
l_ch 0	l_ch 1	pld_flow	length0	length1	length2	length3	length4
data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7
...
data bit 0	data bit 1	data bit 2	data bit 3	data bit 4	data bit 5	data bit 6	data bit 7

n: data length

FIG._25







20 / 45

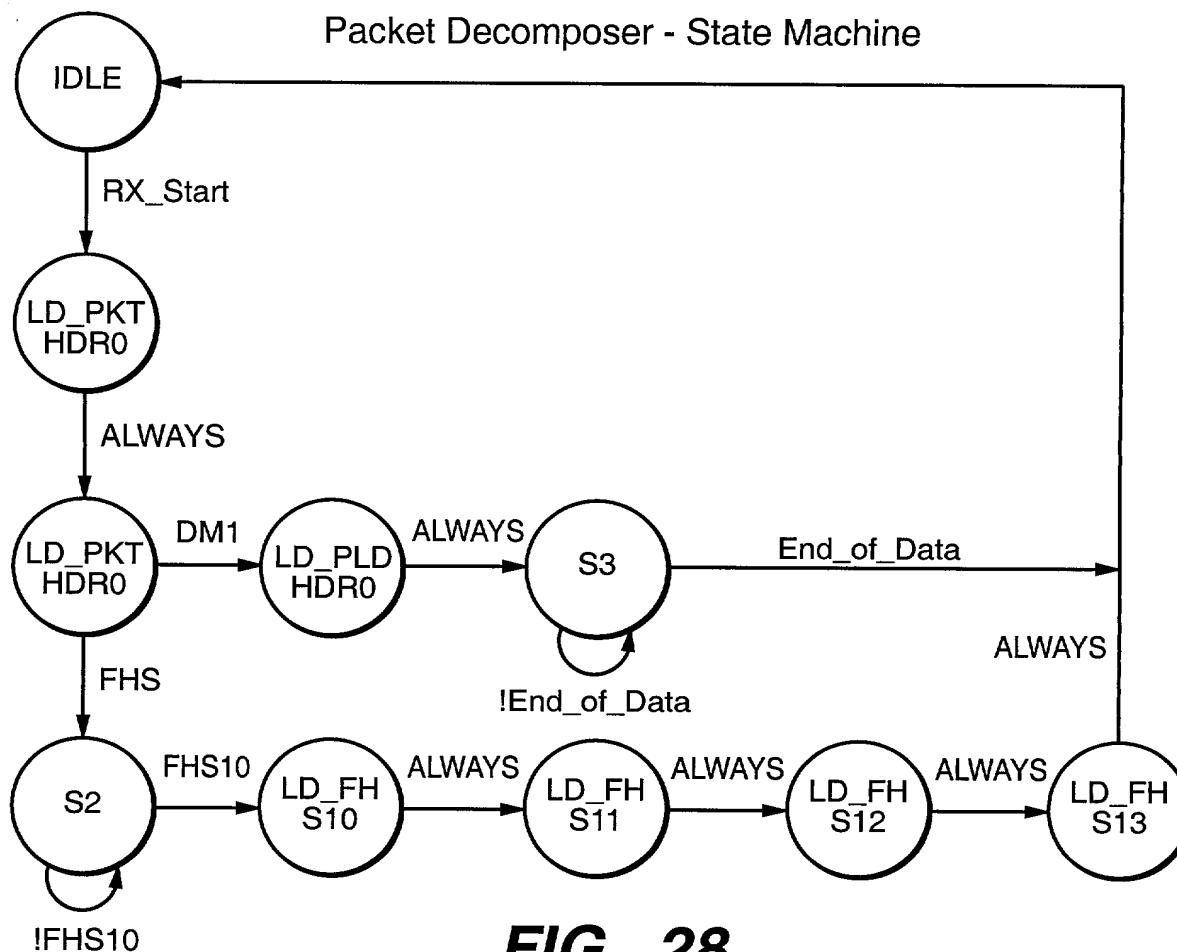


FIG._28

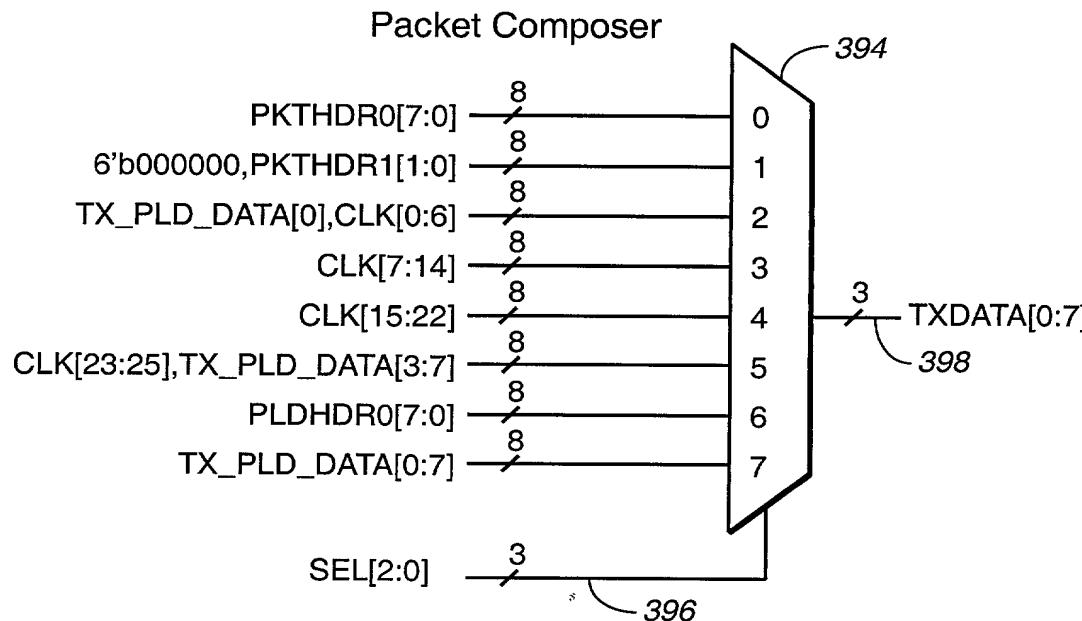


FIG._30



21 / 45

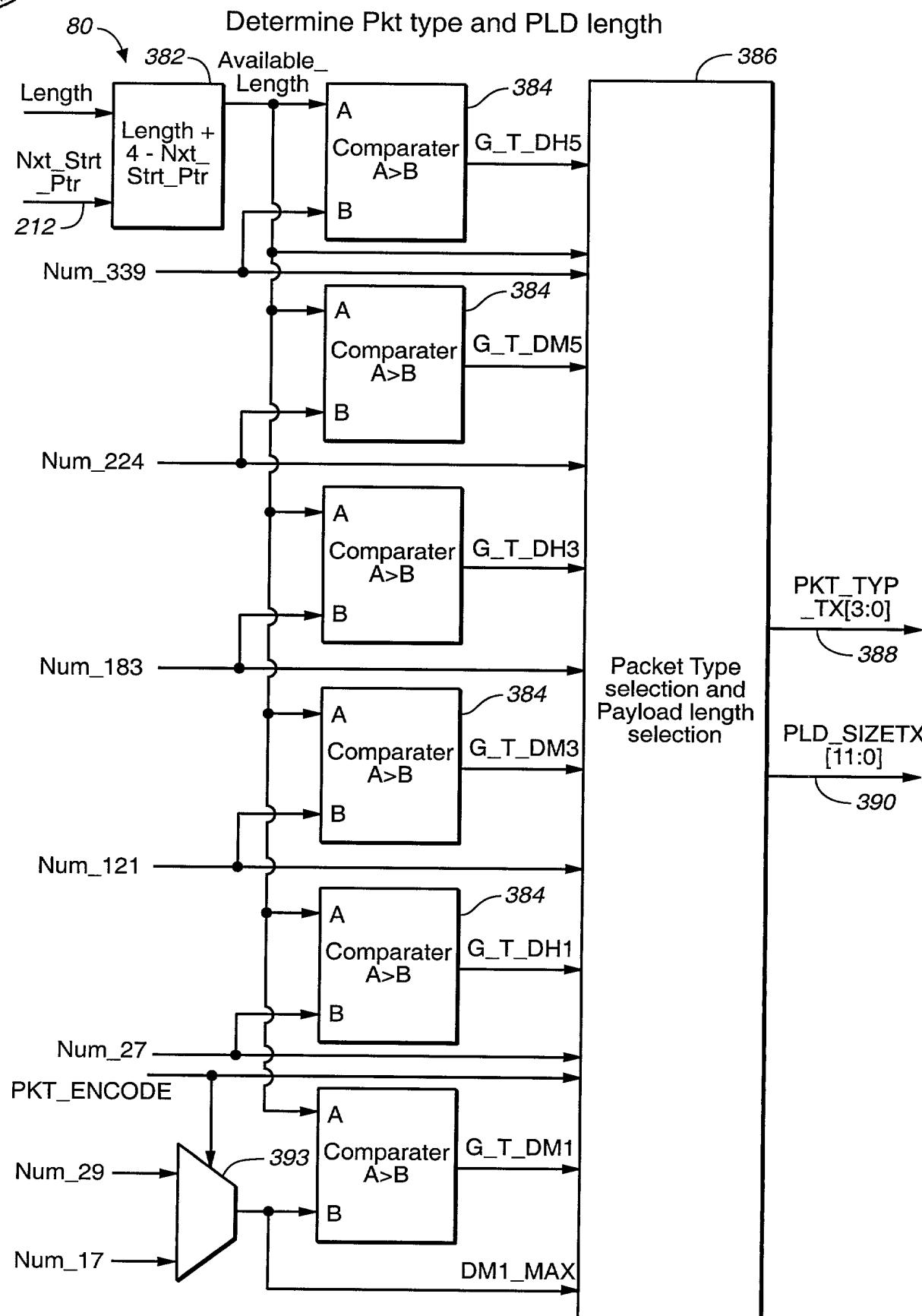


FIG. 29

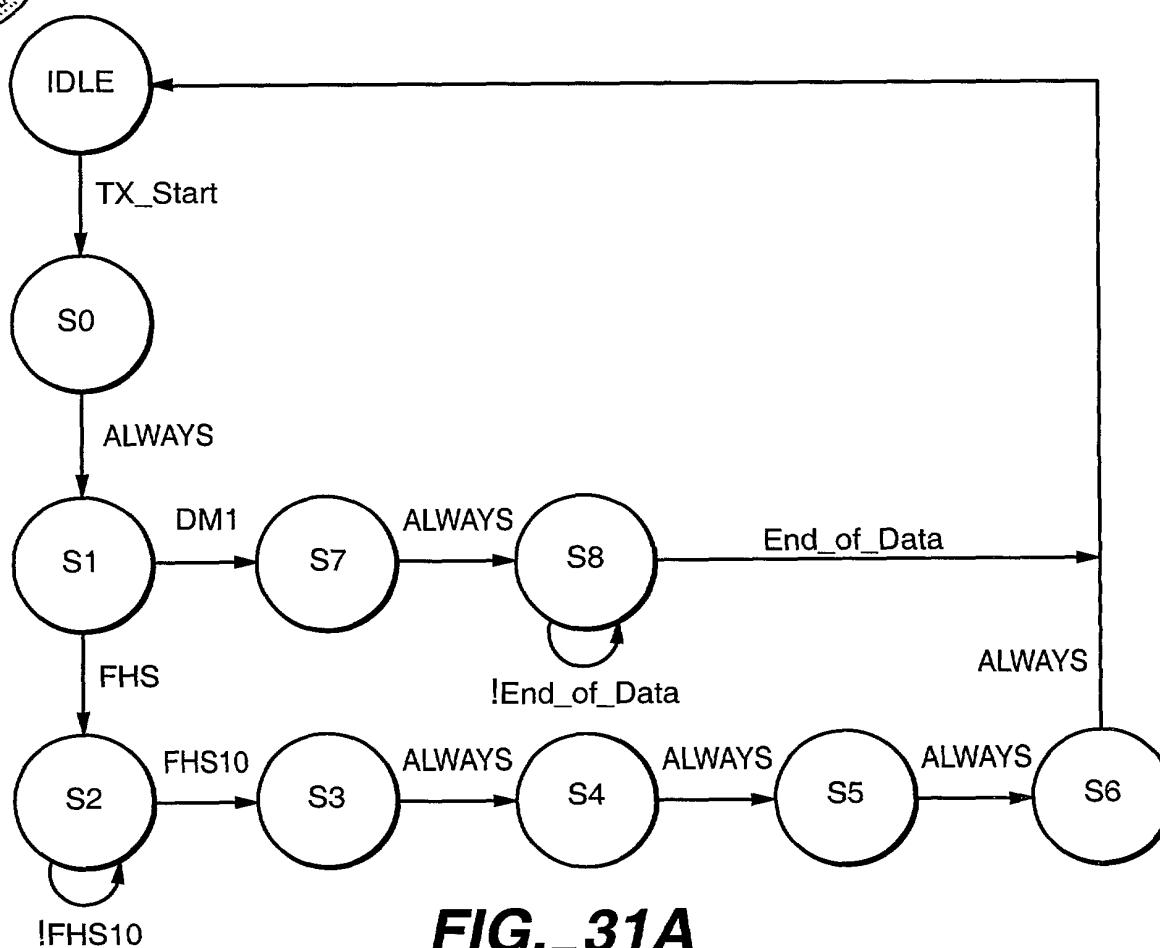


FIG._31A

State	SEL[2:0]
IDLE	0
S0	0
S1	1
S2	7
S3	2
S4	3
S5	4
S6	5
S7	6
S8	7

FIG._31B



Dual pointer buffer control scheme (1)

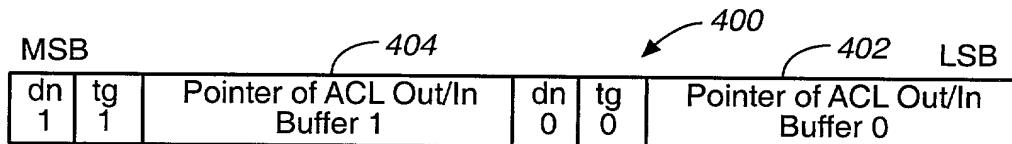


FIG._32

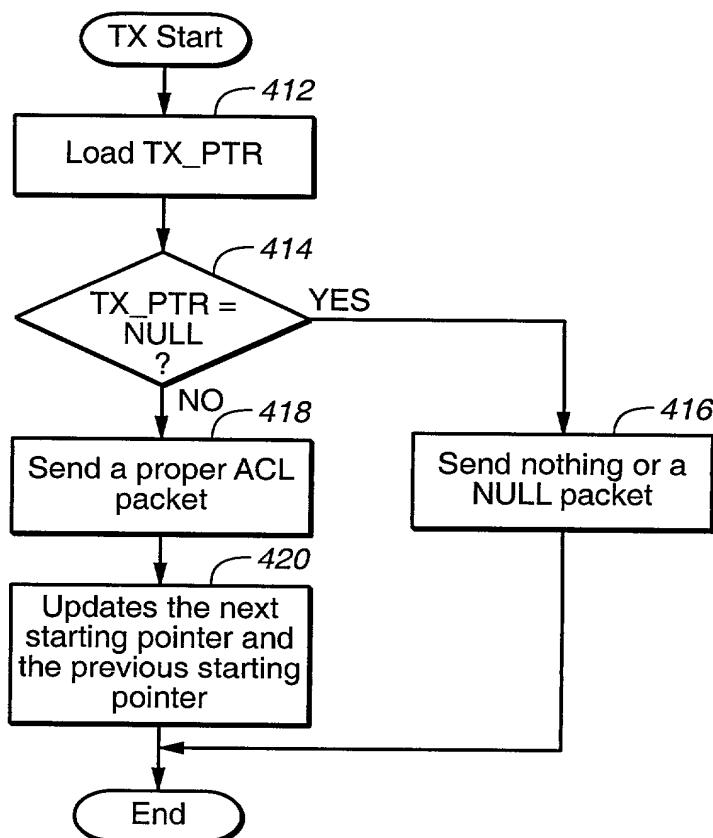


FIG._33



How to convert between the L2CAP packet format and buffer format

L2CAP Packet Format		0		8 7		L2CAP Length	
31	24 23	16 15	16	15	16	15	0
	Channel ID		Data Byte 0
...		
	Data Byte n	

FIG._34A

How to convert between the L2CAP packet format and buffer format

ACL Buffer while Sending		0		8 7		L2CAP Length	
31	24 23	16 15	16	15	16	15	0
Flow	na	Data Total Length	BC	1	0	Fv	Flush Expiration Time[12:2]
	Channel ID		
...		
	Data Byte n	
		
Flsh	na	4				Pkt Encode	4

FIG._34B

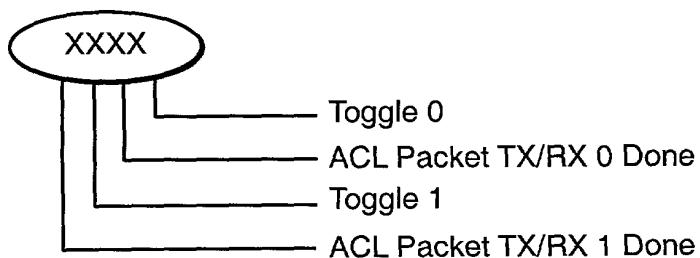
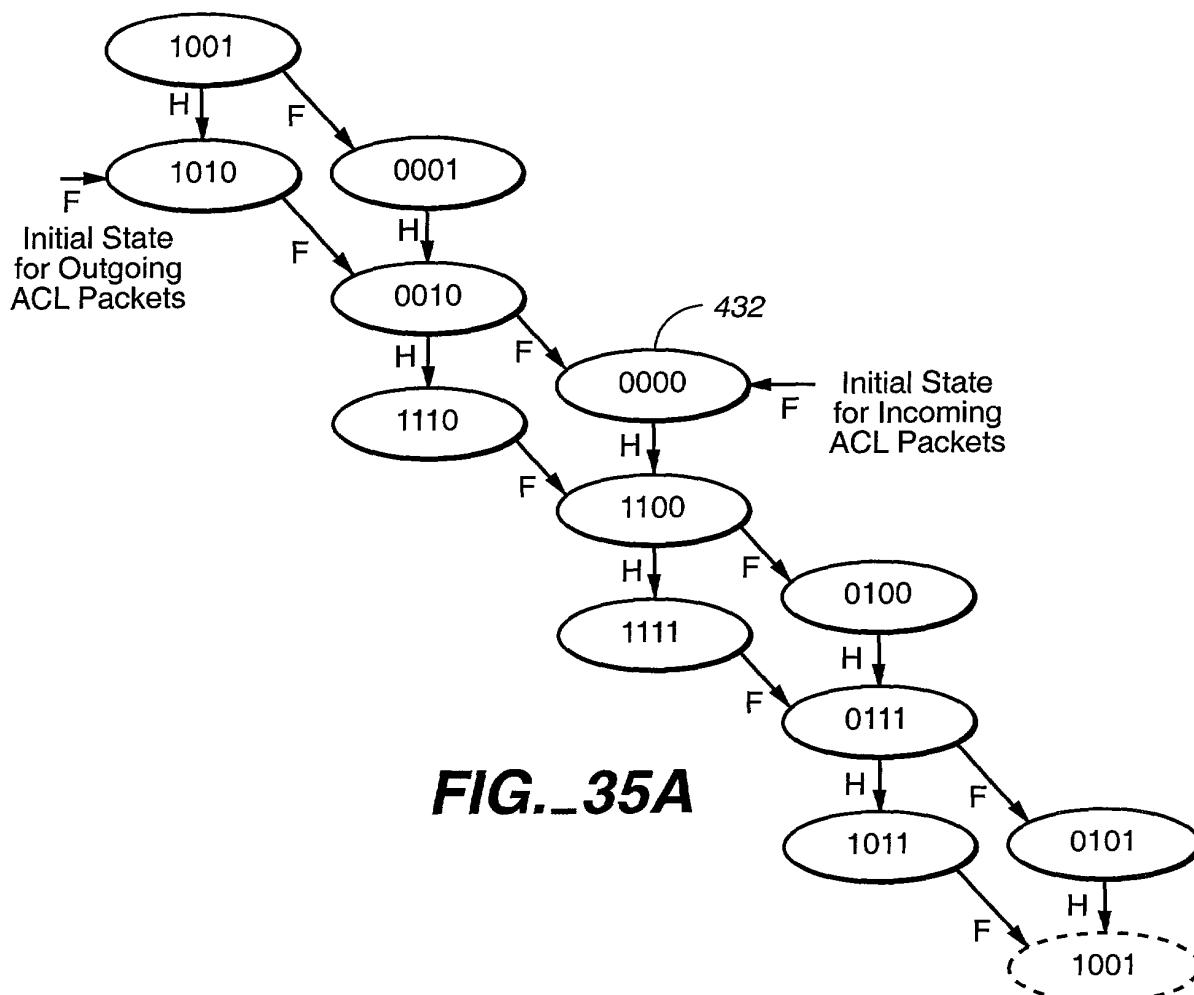


FIG._35B



FIG. 36A

Step 1: After initialization, the value of pointers is "NULL".

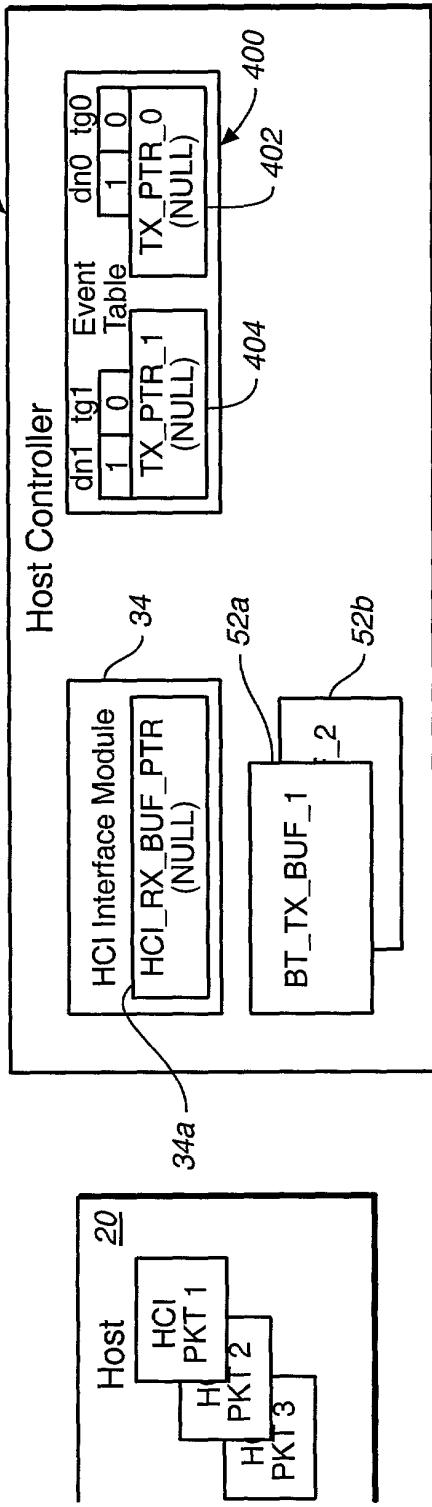


FIG. 36B

Step 2: The Host Controller assigns a buffer 'BT_TX_BUF_1' to receive the HCI packet.

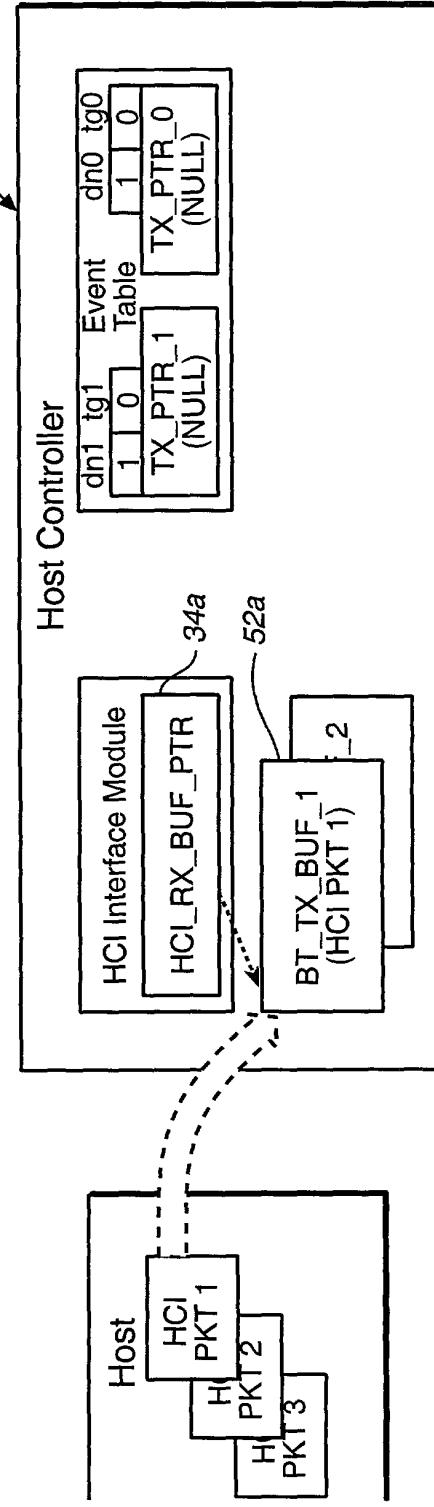




FIG.-36C

Step 3: The HCI packet 'HCI PKT 1' is stored in the buffer 'BT_TX_BUF_1'.

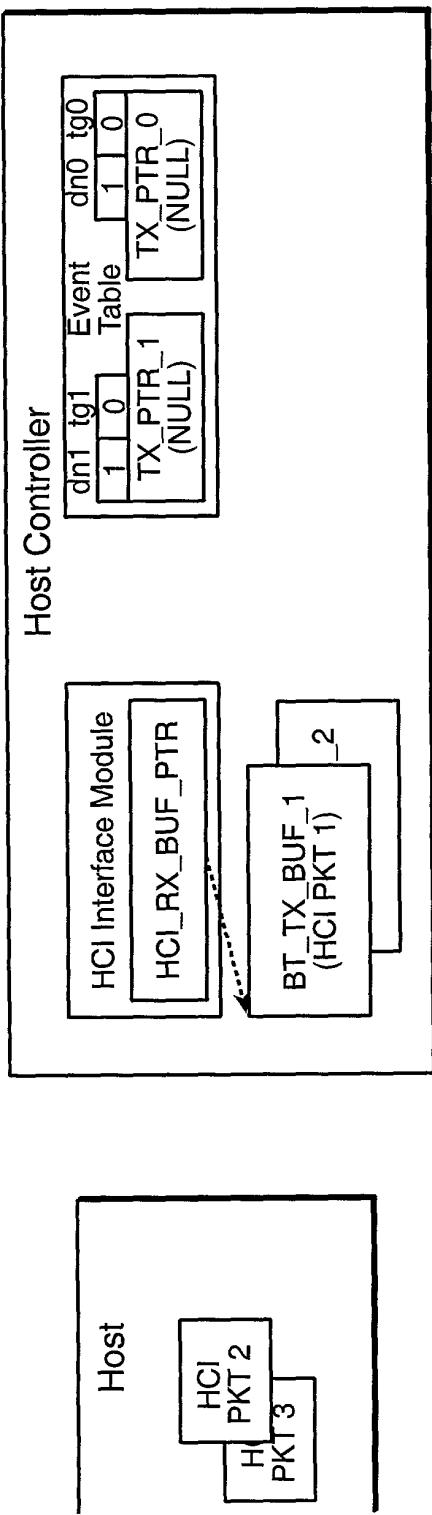
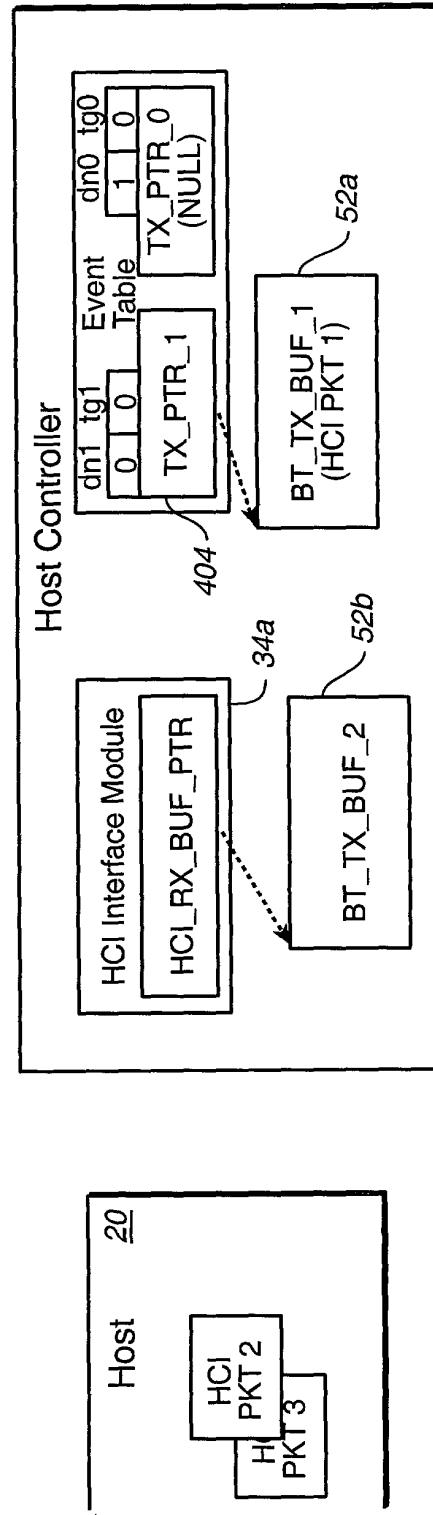


FIG.-36D

Step 4: The Host Controller assigns another buffer 'BT_TX_BUF_2' to receive the HCI packet.



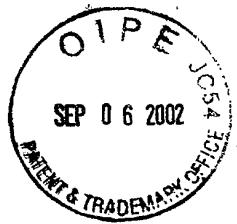


FIG. - 36E

Step 5: While the Bluetooth Module is sending the HCI packet 1, the HCI Interface Module is receiving the HCI packet 2.

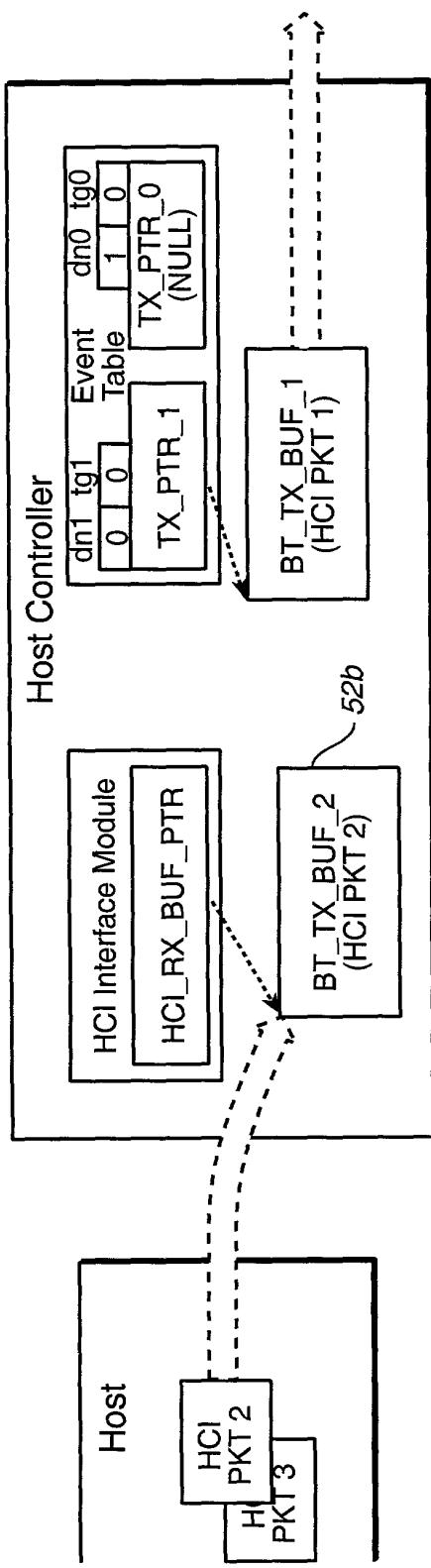
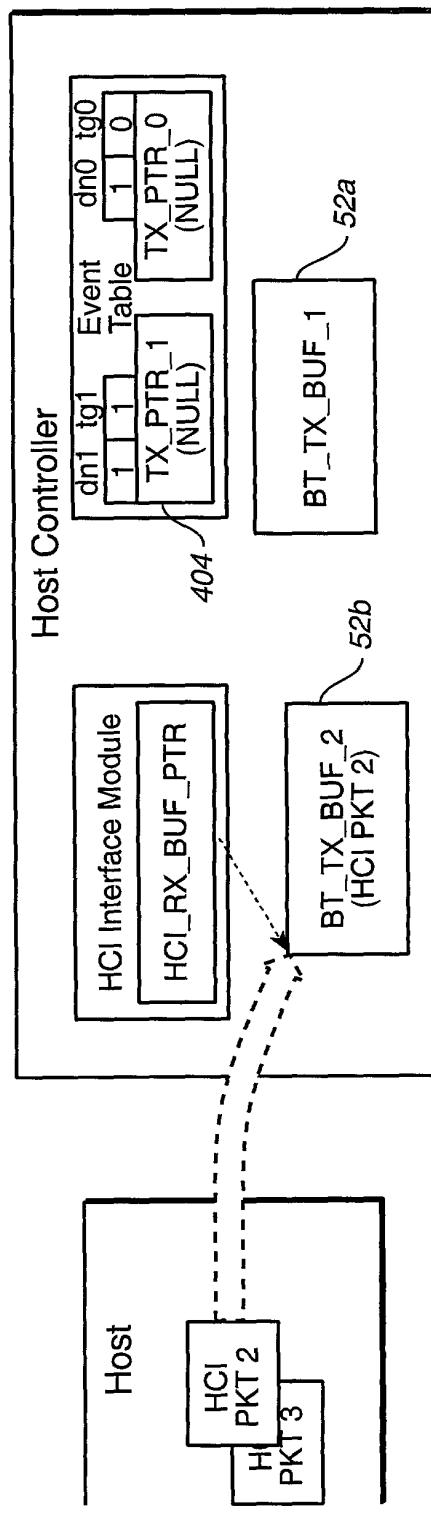


FIG. - 36F

Step 6: Assuming that the HCI packet 1 is sent before the HCI packet 2 is received, the buffer 'BT_TX_BUF_1' is released.





29 / 45

FIG._36G
 Step 7: After the HCI packet 2 has been received, the buffer 'BT_TX_BUF_2' is pointed by 'TX_PTR_0'.

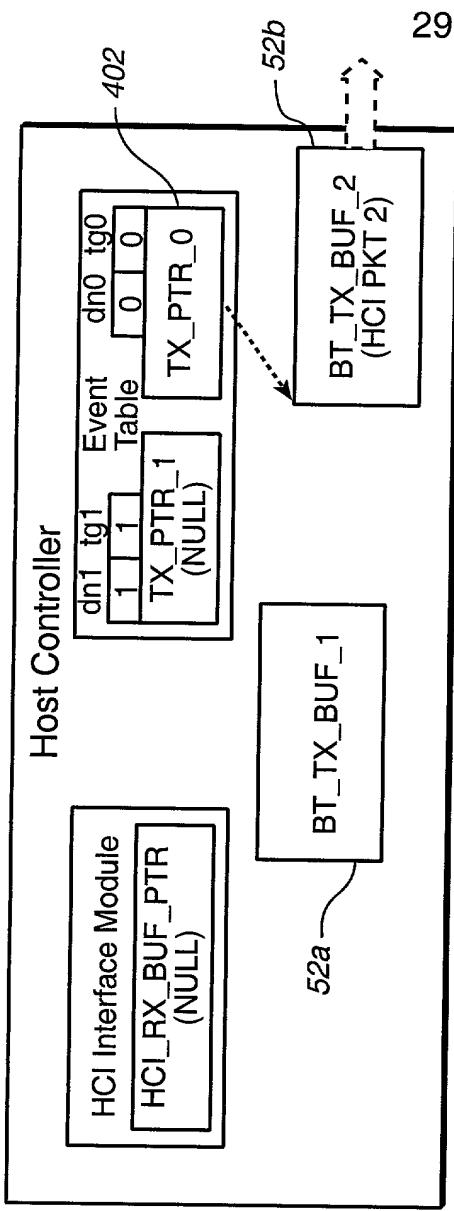


FIG._36H
 Step 8: The free buffer 'BT_TX_BUF_1' is assigned to the HCI Interface Module again to receive another HCI packet.

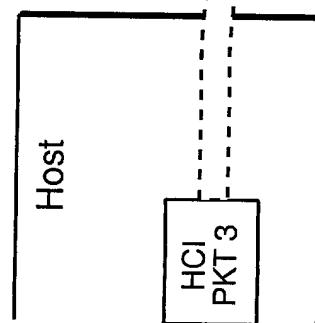
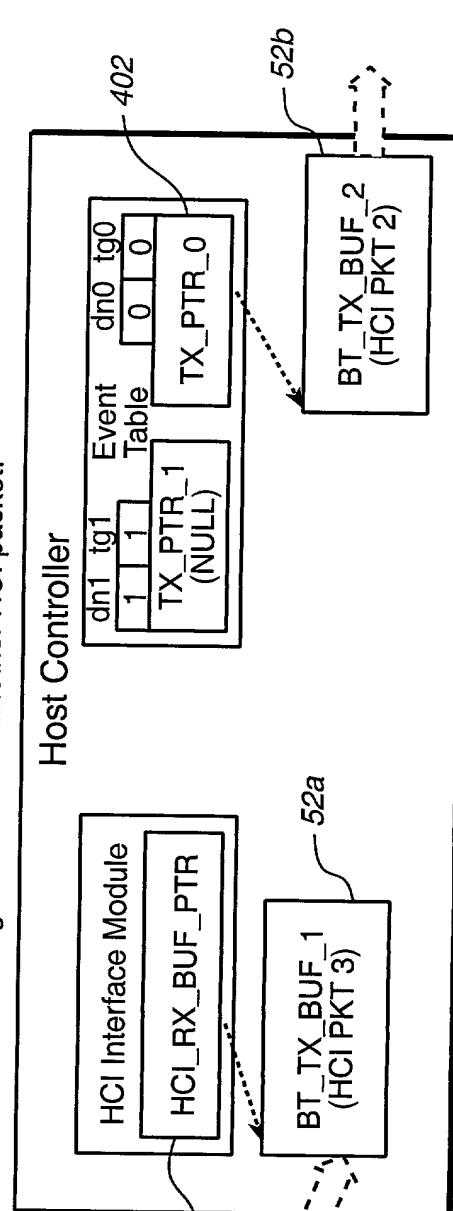
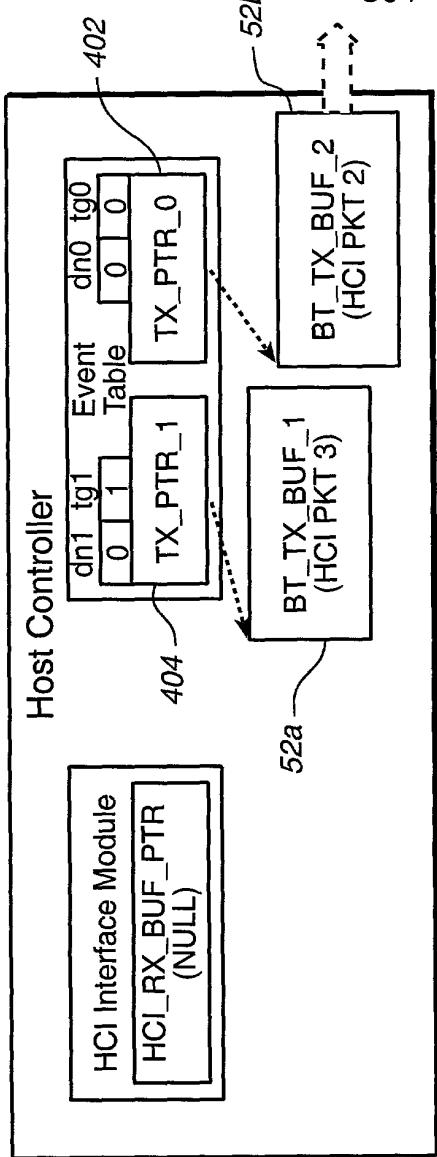




FIG. 36I

Step 9: Assuming that the HCI packet 3 is received before the HCI packet 2 is sent, 'TX_PTR_1' points to buffer 'BT_TX_BUF_1'.



30 / 45

FIG. 36J

Step 10: The free buffer 'BT_TX_BUF_2' is assigned to the HCI Interface Module again to receive another HCI packet.

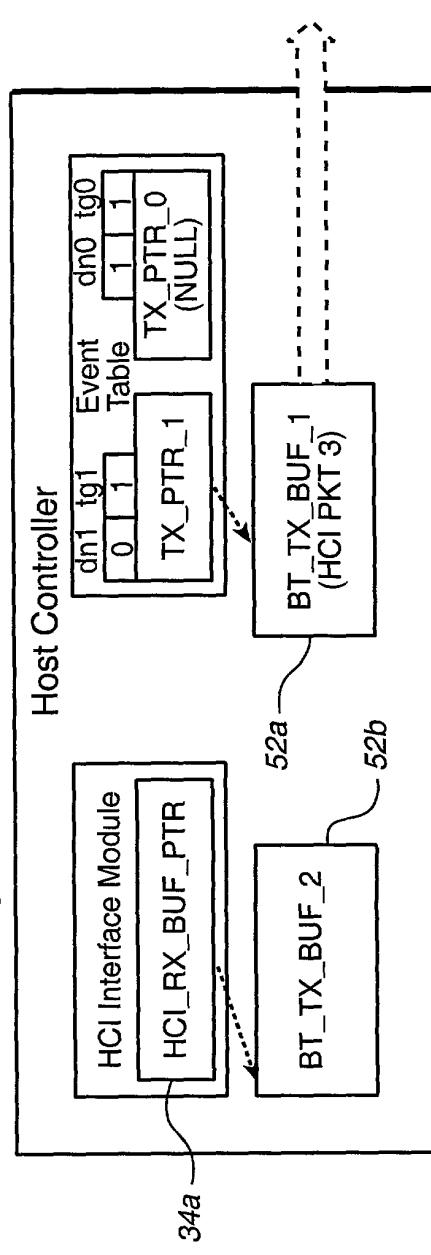




FIG.-36K

Step11: After the HCI packet 3 has been transmitted, the buffer 'BT_TX_BUF_1' will be released.

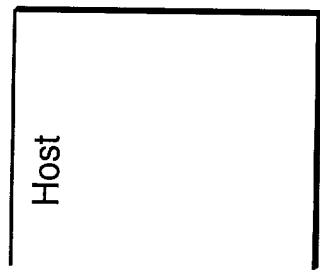
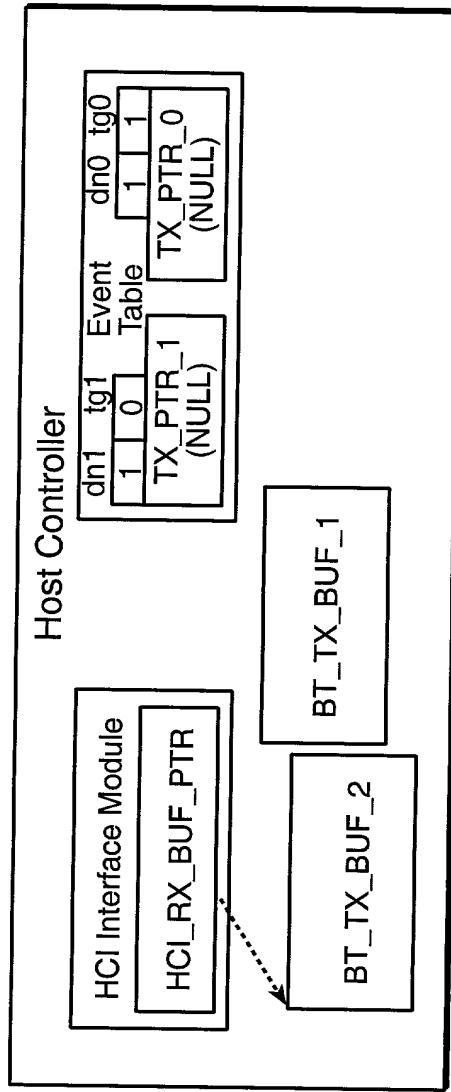




FIG._37A

Step 1: After initialization, the value of pointers is "NULL". Assuming that two receiving buffer are available.

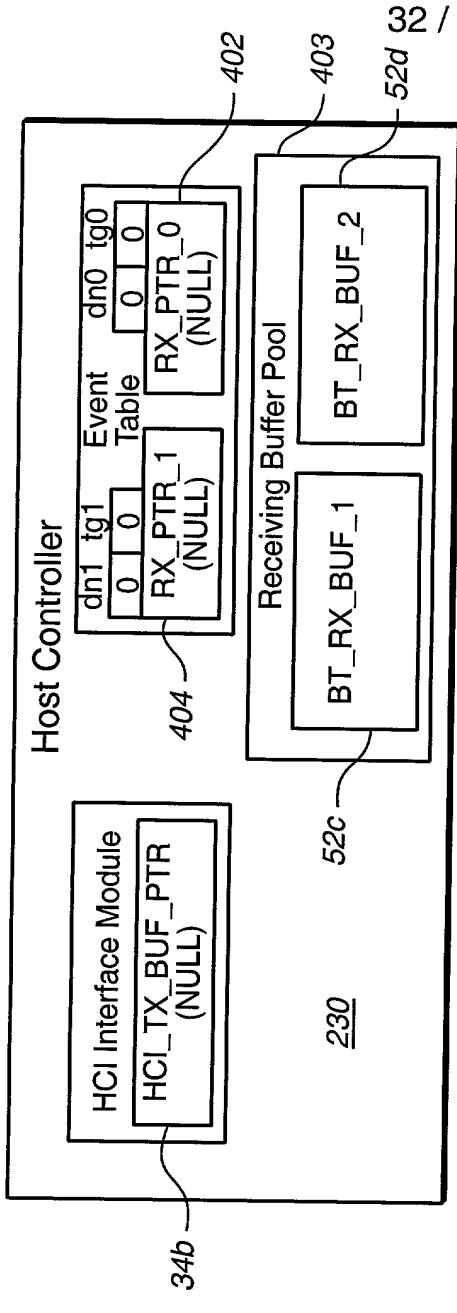


FIG._37B

Step 2: The Bluetooth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.

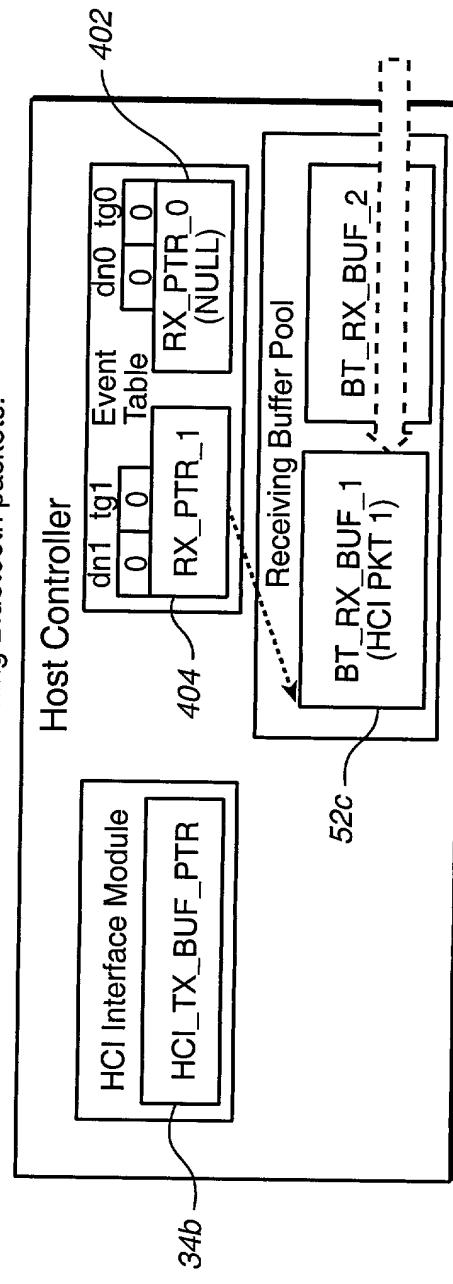




FIG.-37C

Step 3: The buffer 'BT_RX_BUF_1' is released when any one of the three buffer releasing conditions is detected.

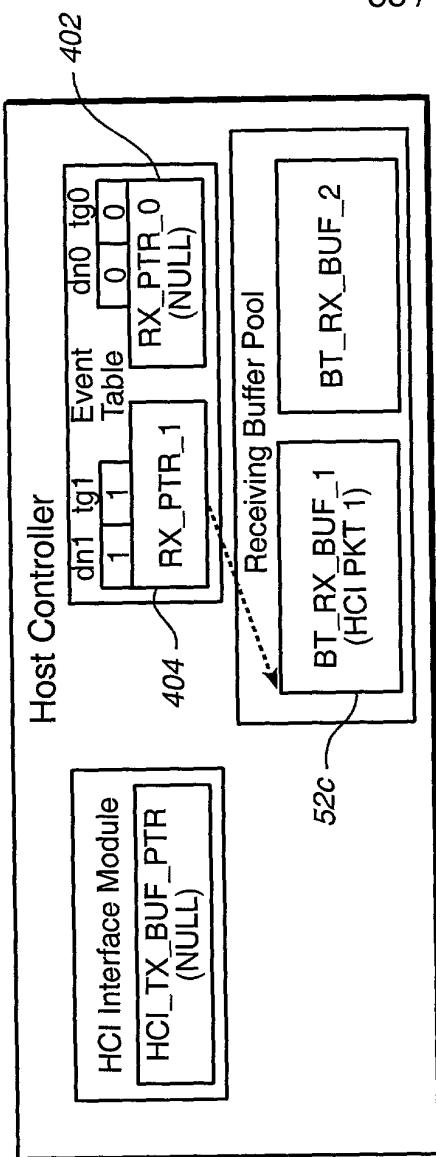


FIG.-37D

Step 4: Firmware releases this buffer 'BT_RX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.

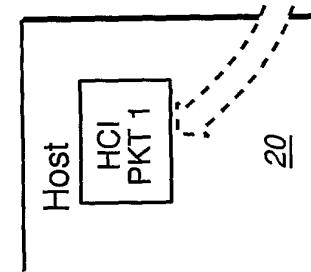
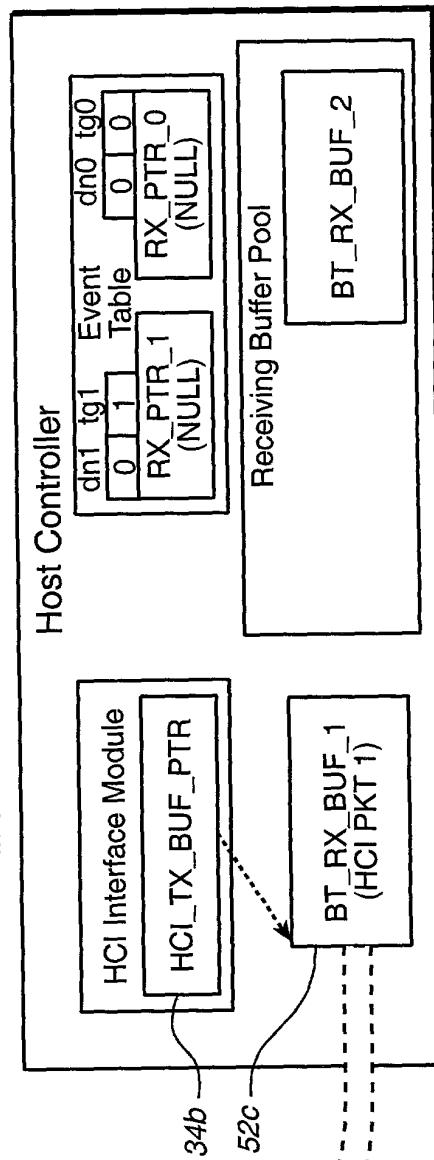
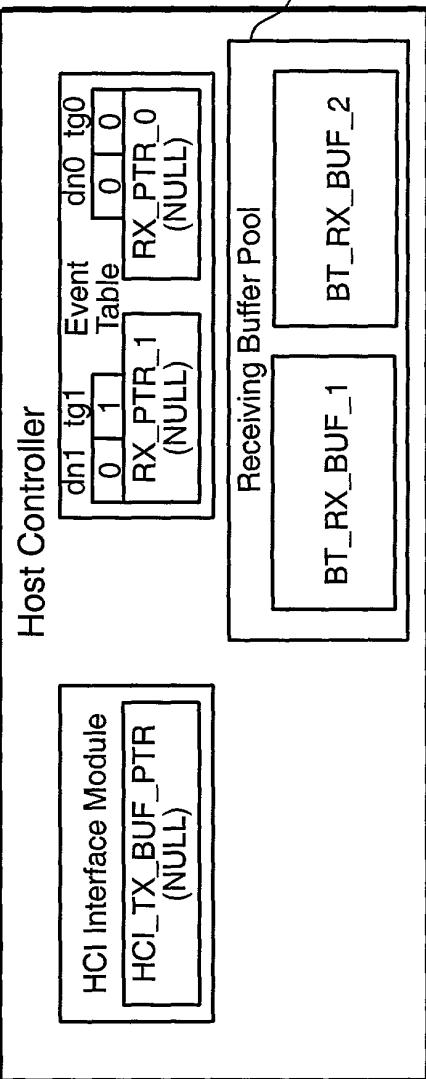
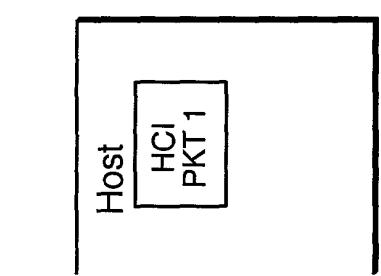




FIG._37E

Step 5: After the HCI packet 1 is sent to the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool.



34 / 45

FIG._37F

Step 6: The Bluetooth Module assigns buffer 'BT_RX_BUF_1' to receive the incoming Bluetooth packets.

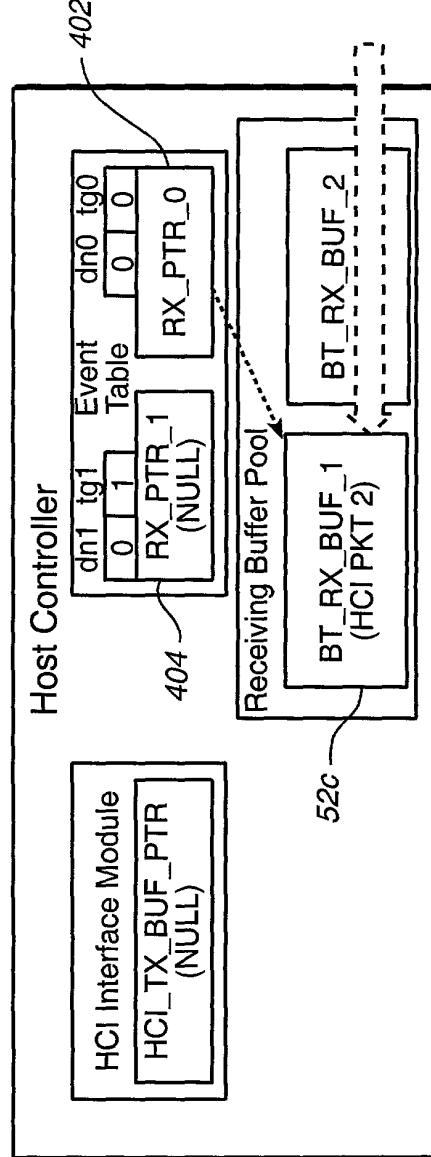
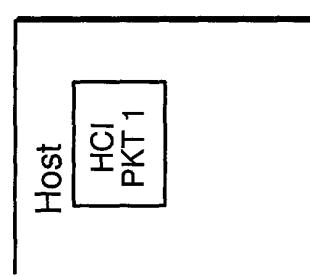
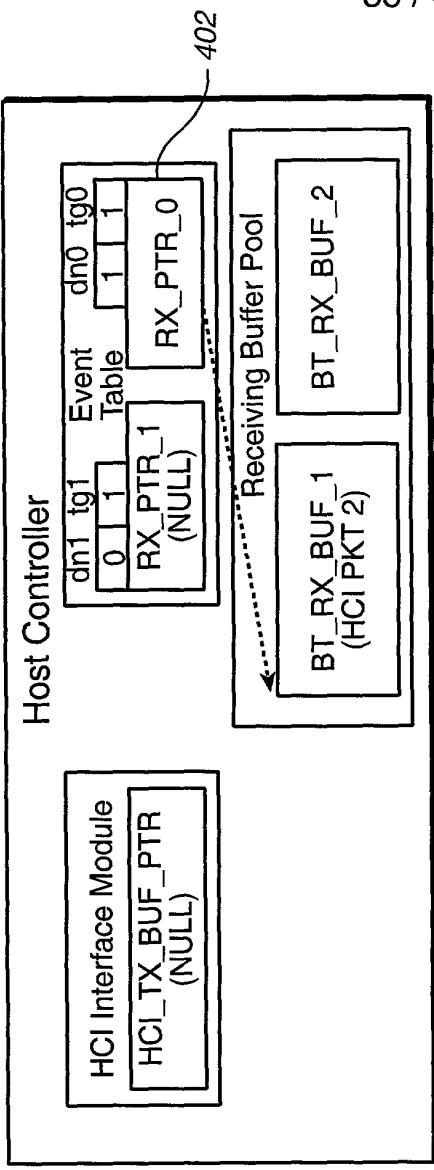


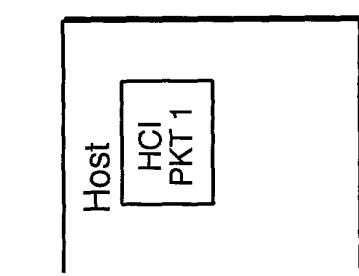


FIG._37G



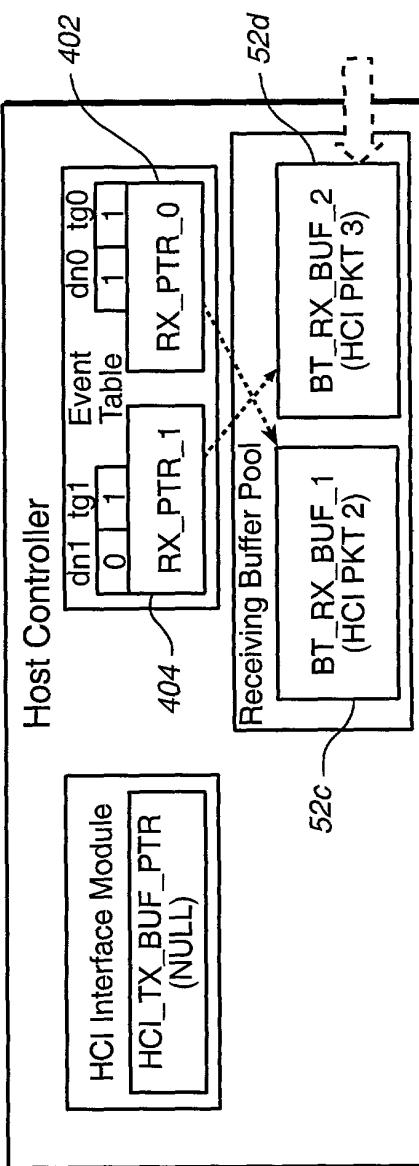
Step 7: The buffer 'BT_RX_BUF_1' released when any one of the three buffer releasing conditions is detected.

FIG._37H



Step 8: Before buffer 'BT_RX_BUF_1' removed by the firmware, another buffer is assigned to receive data.

Step 7: The buffer 'BT_RX_BUF_1' released when any one of the three buffer releasing conditions is detected.



Step 8: Before buffer 'BT_RX_BUF_1' removed by the firmware, another buffer is assigned to receive data.



FIG._37I

Step 9: Firmware releases this buffer 'BT_TX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.

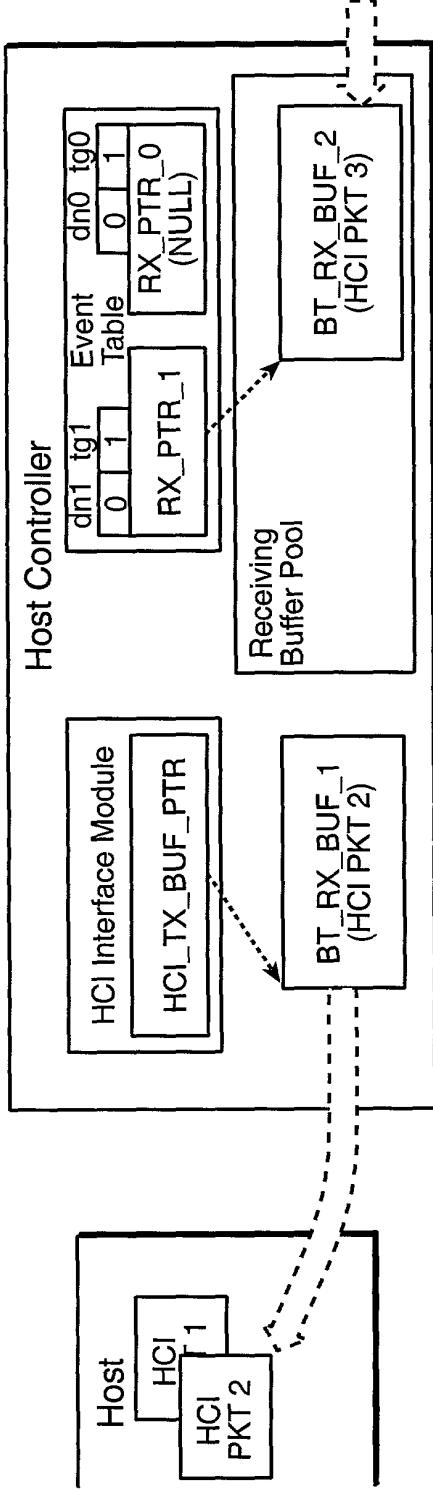


FIG._37J

Step 10: The buffer 'BT_TX_BUF_2' is released when any one of the three buffer releasing conditions is detected.

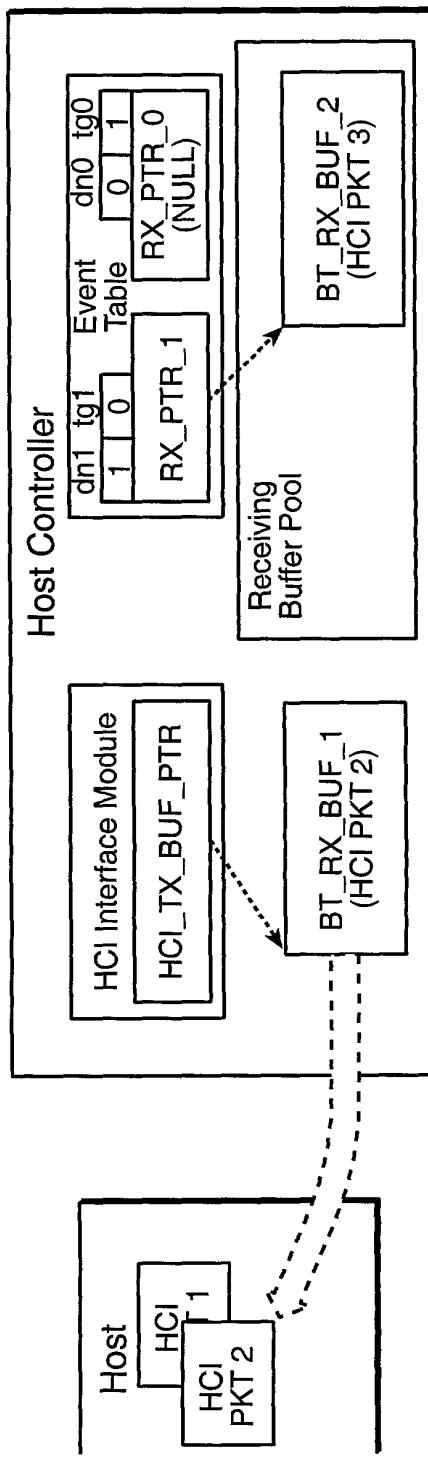
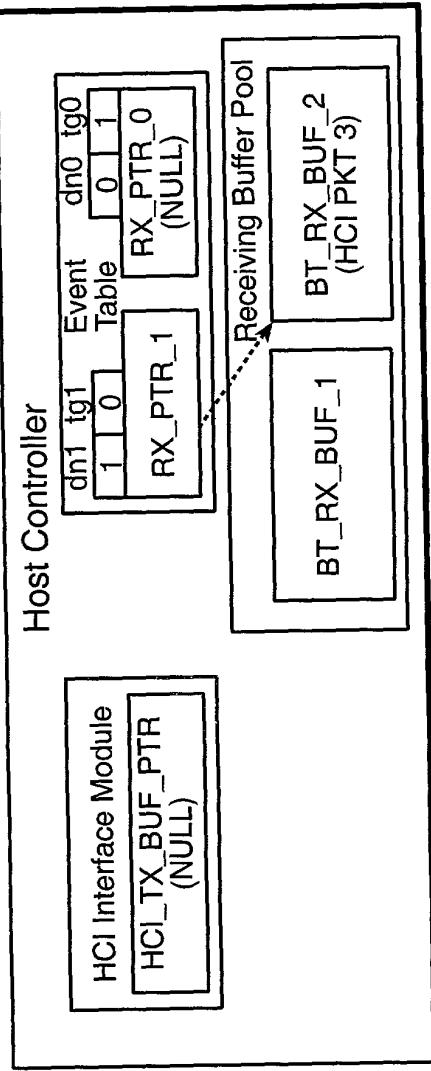


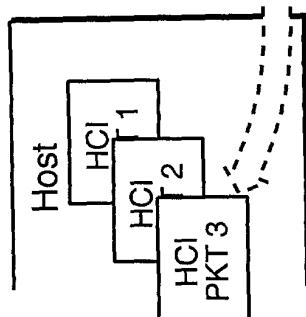


FIG..-37K

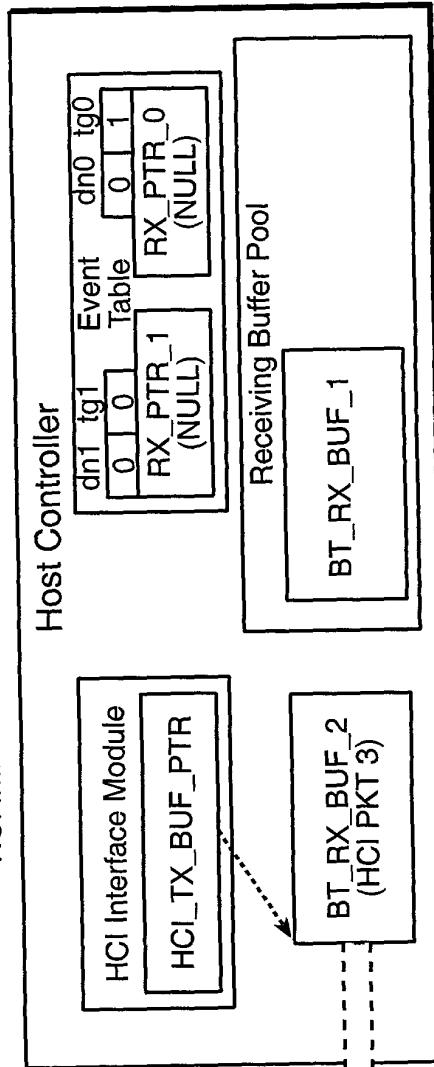


Step 11: After the HCI packet 1 is sent the Host, buffer 'BT_RX_BUF_1' is released and put back to the receiving buffer pool.

FIG..-37L



Step 12: Firmware releases this buffer 'BT_RX_BUF_1' and sends it to the HCI Interface Module. Then sets the done bit to 0.





38 / 45

Dua Pointer Buffer Scheme 1: Hardware implementation

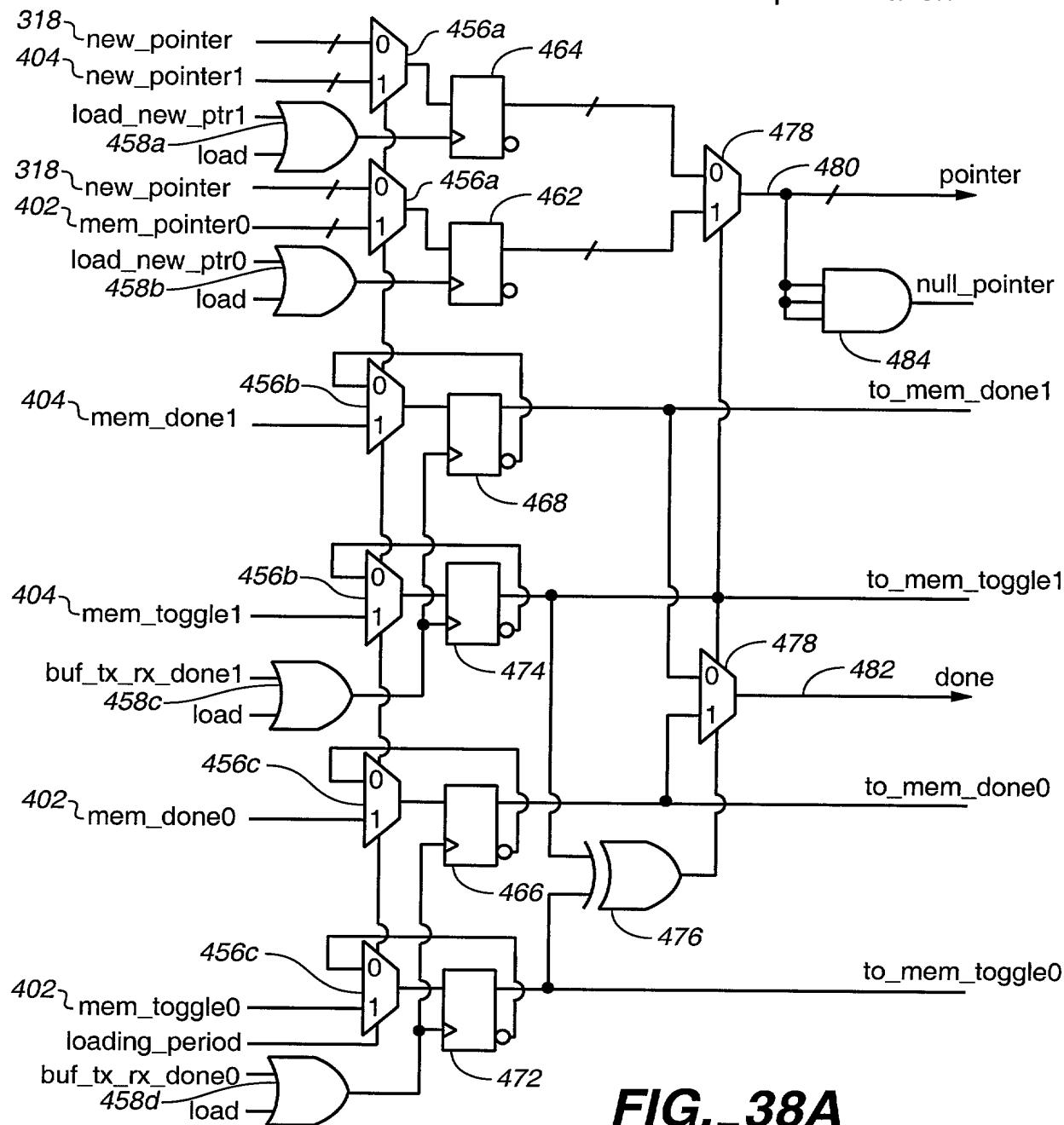


FIG._38A

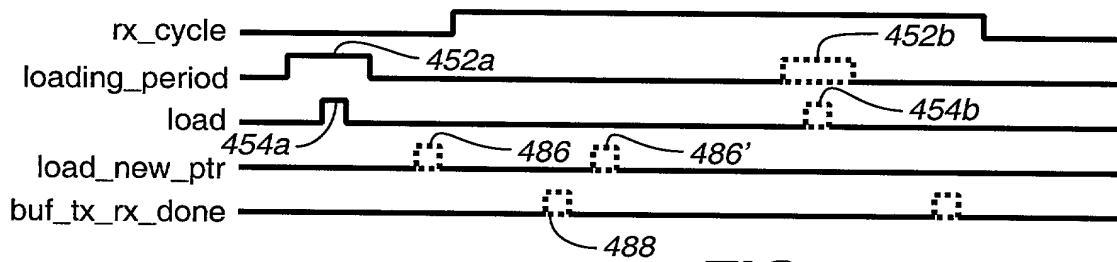


FIG._38B



Duel Pointer Buffer Scheme 1: Firmware implementation

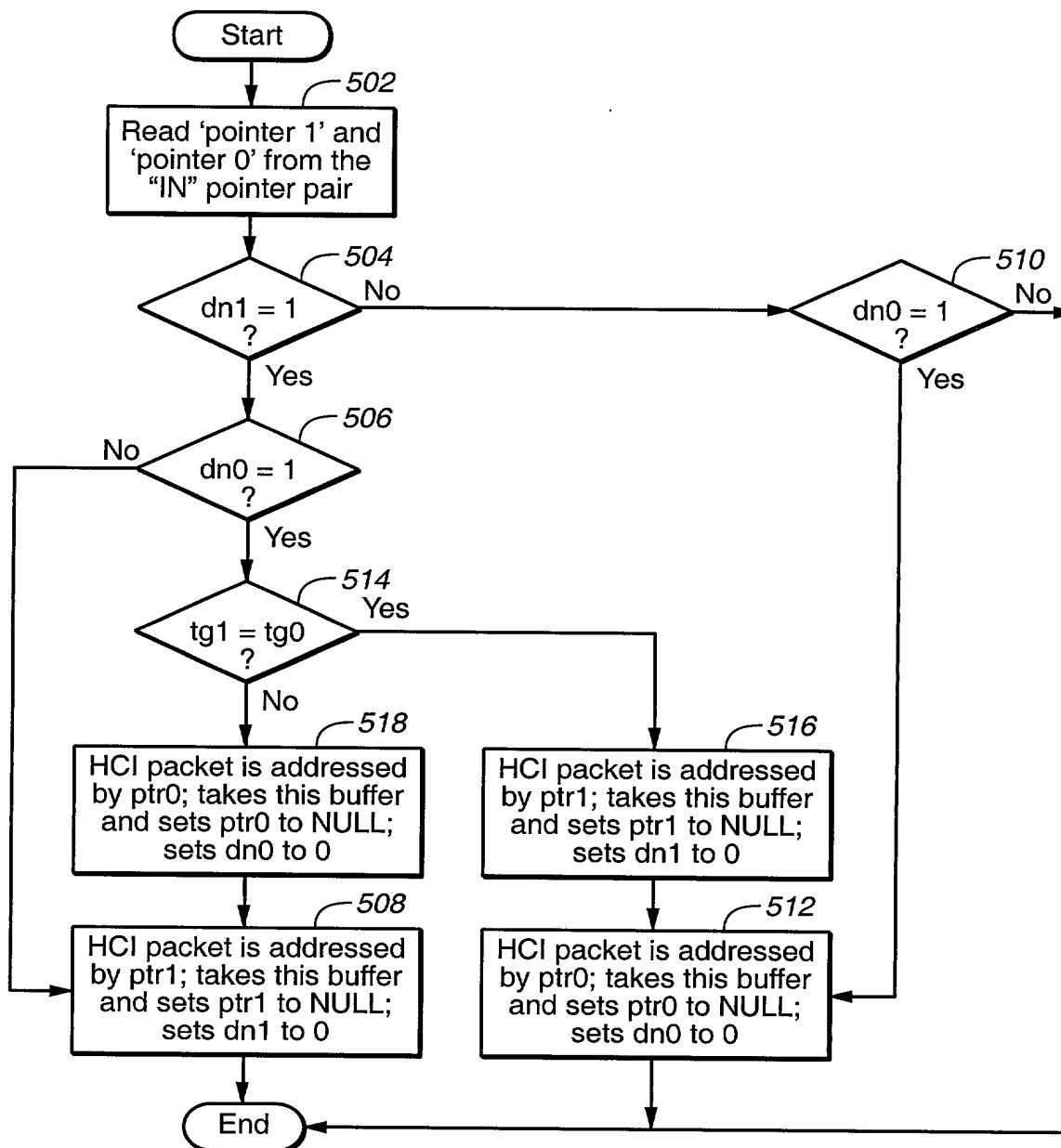


FIG._39



Duel Pointer Buffer Scheme 1: Firmware implementation

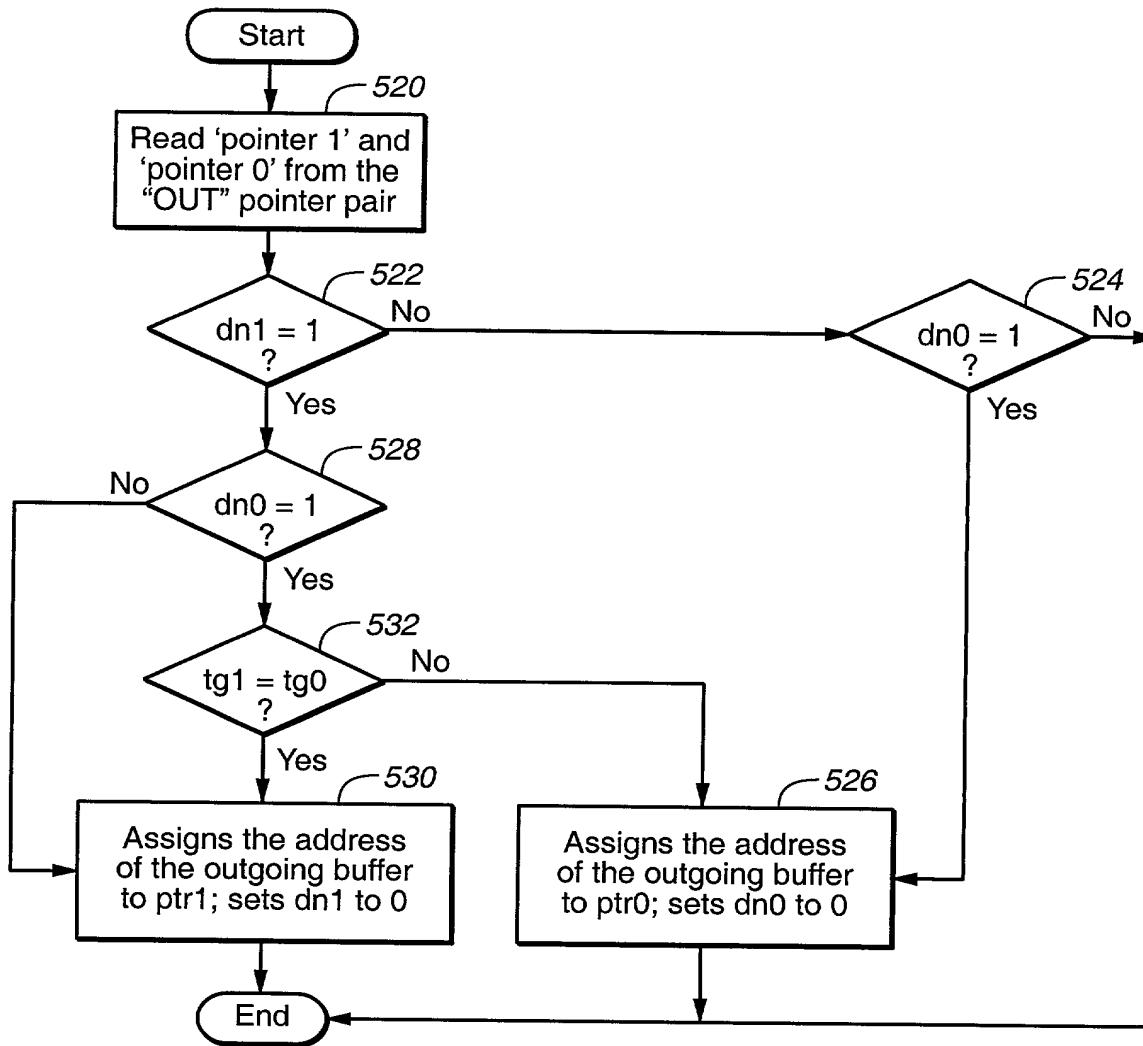


FIG._40



Duel Pointer Buffer Scheme 1: Firmware implementation

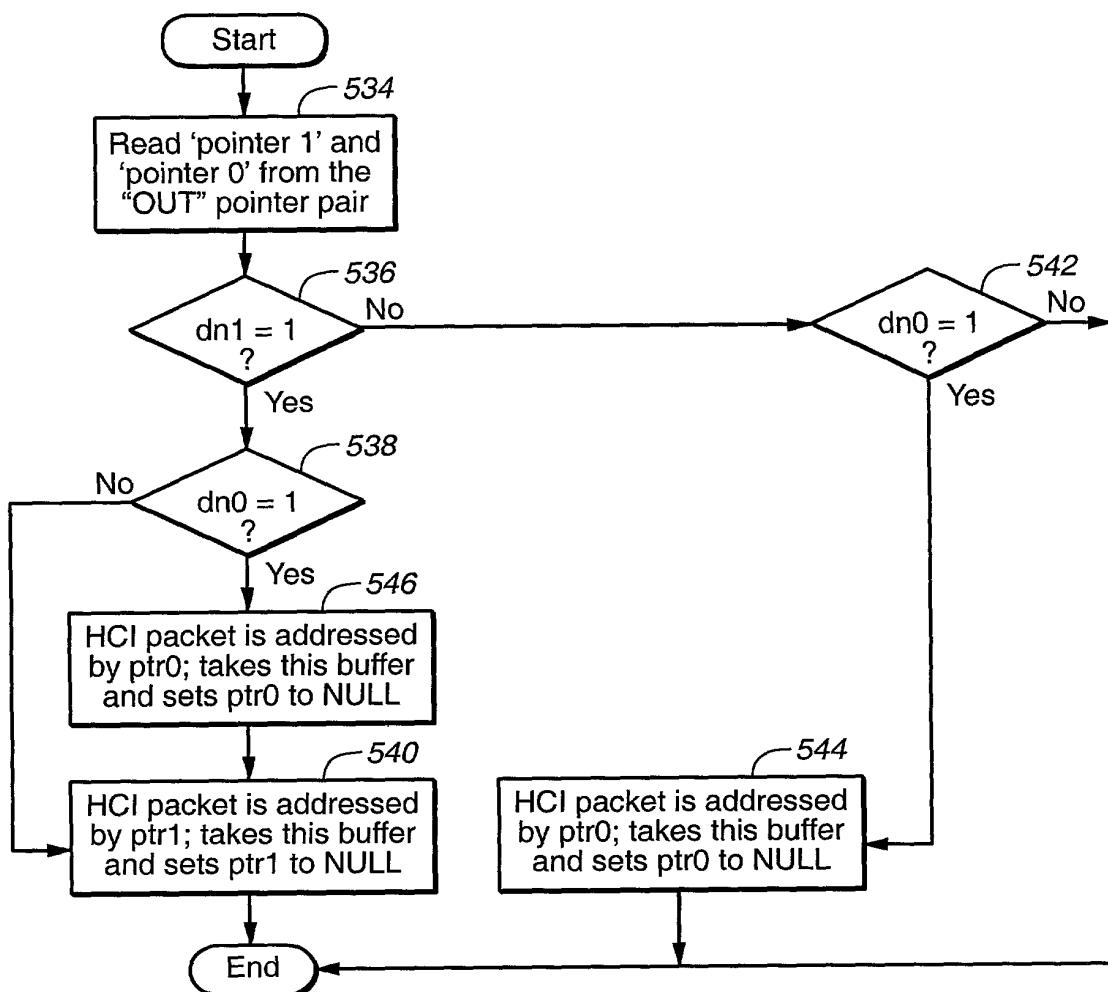


FIG..41

42 / 45

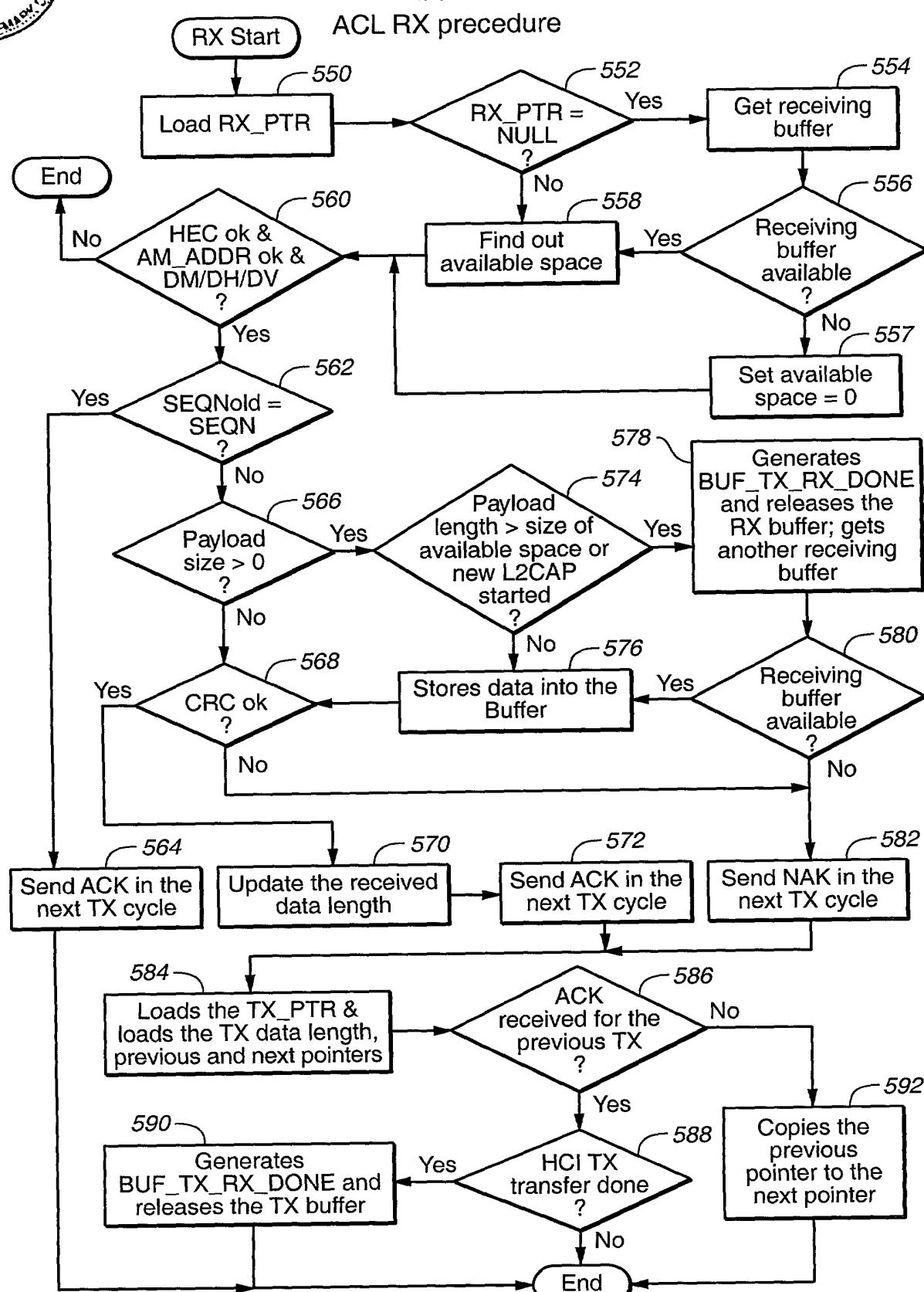


FIG. 42

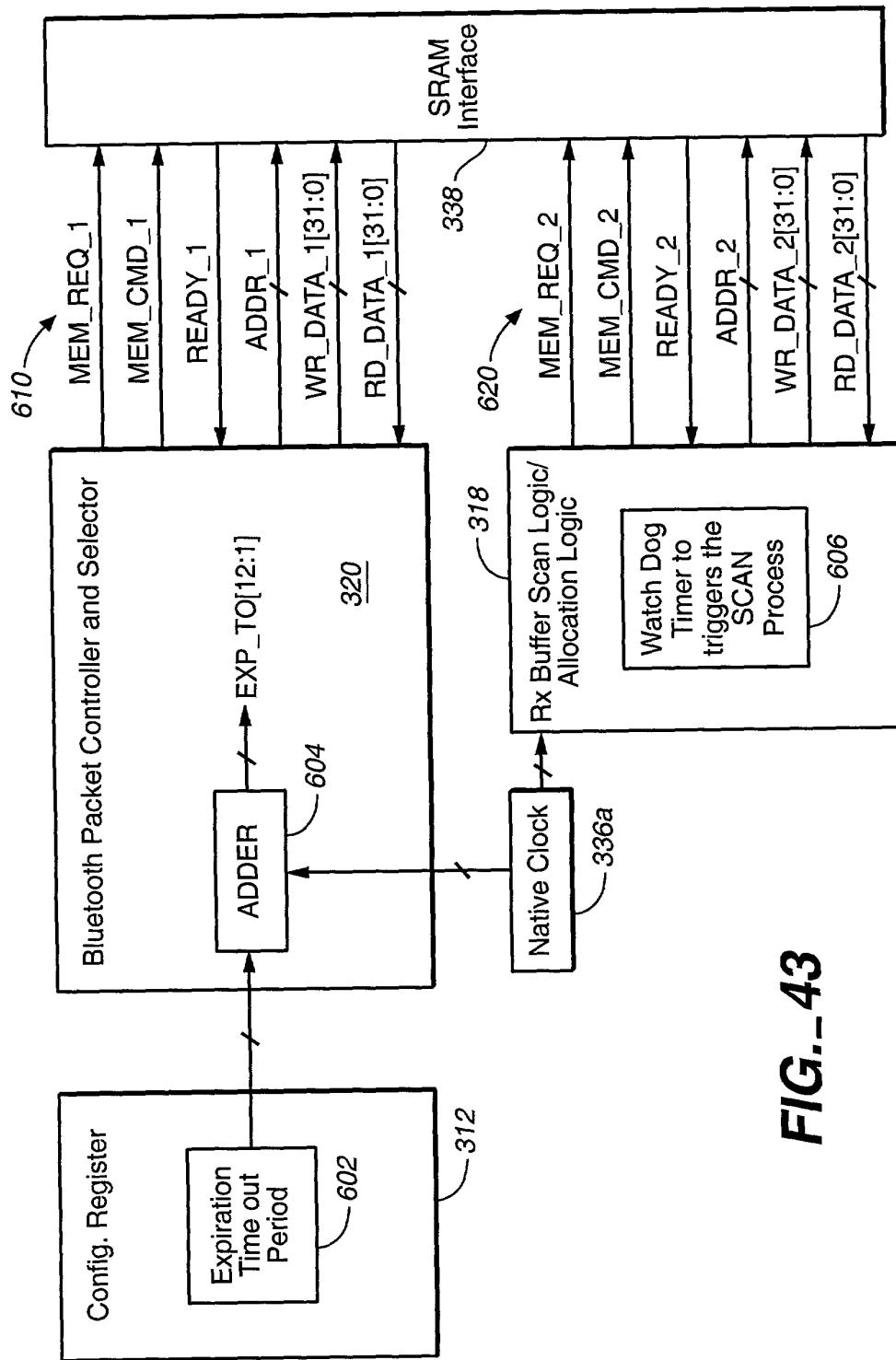


FIG. - 43

ACL RX Buffer is released due to expiration time-out

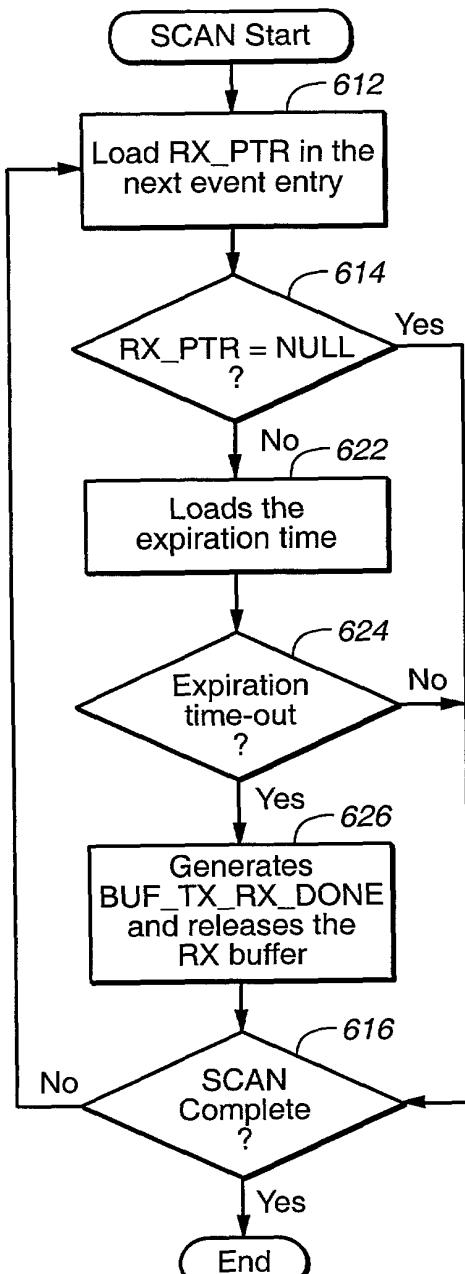


FIG. 44

Scheduler - Scheduling priority of ACL links in master mode

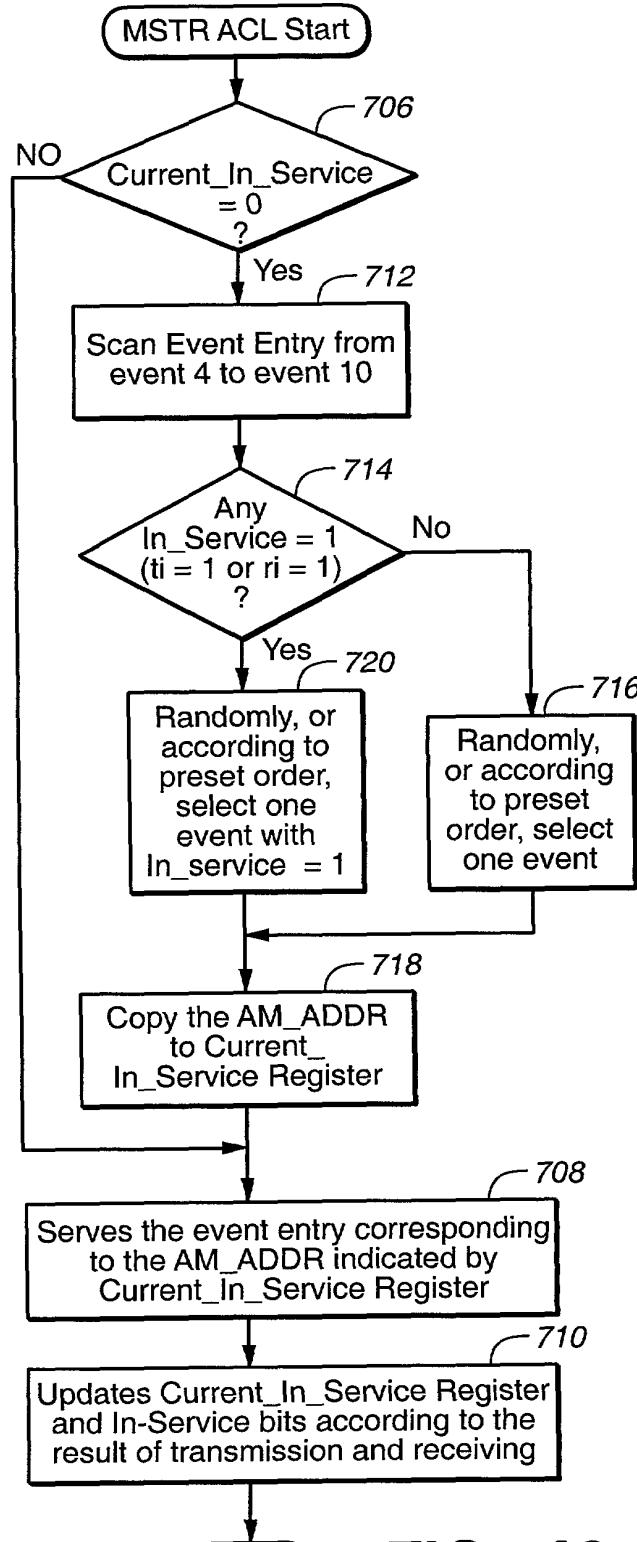


FIG. 46

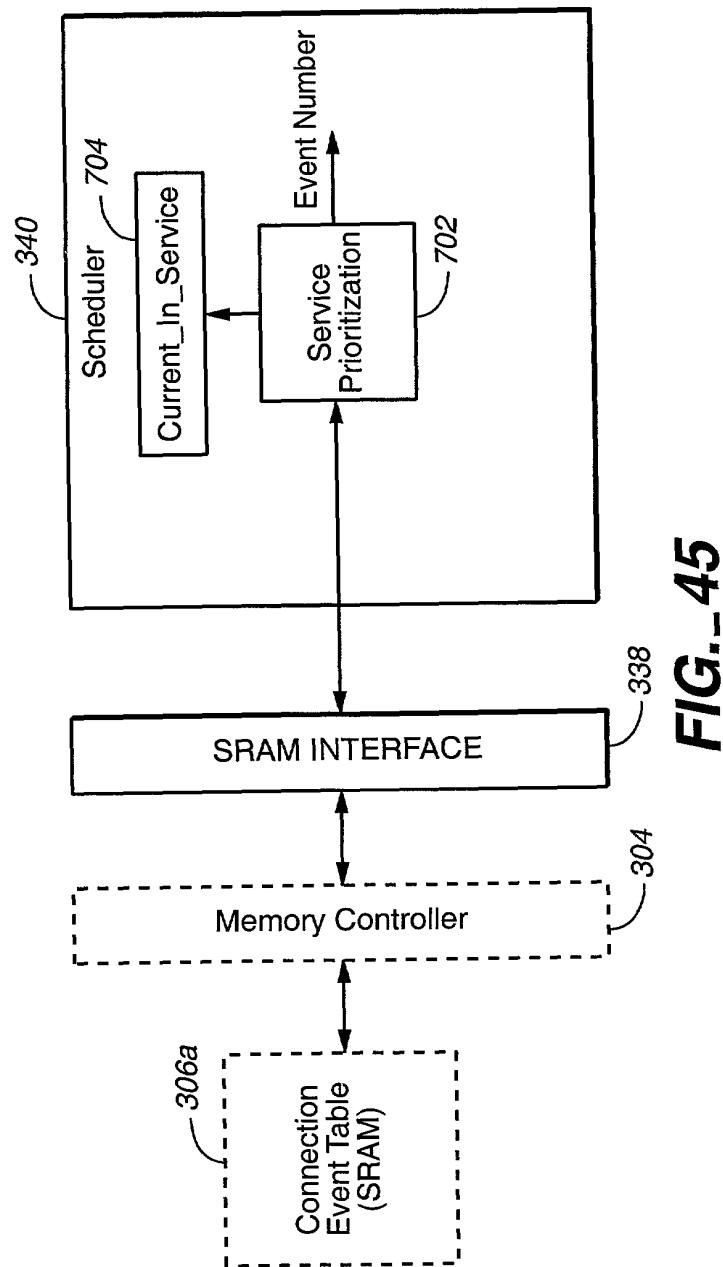


FIG. 45